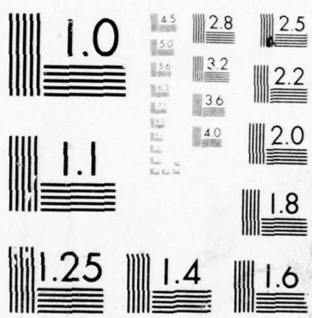
AD-A069 896 FLORIDA UNIV EGLIN AFB GRADUATE ENGINEERING CENTER F/G 13/13
FAILURE OF UNDERGROUND HARDENED STRUCTURES SUBJECTED TO BLAST L--ETC(U)
APR 79 C A ROSS, C C SCHAUBLE AFOSR-78-3592
UNCLASSIFIED AFOSR-TR-79-0679 NL

1 OF 3
AD
A069896


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

13

FAILURE OF UNDERGROUND HARDENED STRUCTURES SUBJECTED TO BLAST LOADING

FINAL REPORT

C. A. ROSS
C. C. SCHAUBLE

APRIL 30, 1979

U.S. AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

GRANT NUMBER: 78-3592

UNIVERSITY OF FLORIDA

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR-79-0679	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Failure of Underground Hardened Structures Subjected to Blast Loading.		5. TYPE OF REPORT & PERIOD COVERED Final - March 31, 1978 - March 30, 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) C. A. Ross, C. C. Schauble		8. CONTRACT OR GRANT NUMBER(s) AFOSR-78-3592
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Florida Graduate Engineering Center, P. O. Box 1918, Eglin AFB, FL 32542		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2301/A6, 62602F A6
11. CONTROLLING OFFICE NAME AND ADDRESS US Air Force Office of Scientific Research/NP, Bolling AFB, D. C. 20332		12. REPORT DATE April 30, 1979
		13. NUMBER OF PAGES 184
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Final rept. 31 Mar 78-30 Mar 79 197p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited; approved for public release

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reinforced concrete
Plastic hinge response
Blast loading
Beams
Plates

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The major objective of this study was to apply plastic hinge response to buried reinforced concrete beams and plates. Using the plastic hinge response, of both stationary and moving hinges, equations of motion were derived for beams and plates using time and spatially varying loading functions. The spatial variations were evaluated in closed form and the time variations were evaluated numerically. The solution of the equations of motion

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407 525

ITEM NO. 20, Continued

were coded in a rather lengthy but efficient computer model with running times on the order of a second for each case. Inconsistencies in predicting pressure loadings in soil, neglecting elastic response, variations in structural designs, experimental errors, etc. could produce as much as +50% variation when comparing analyses to experiments. However, comparison of analyses to limited experimental data showed reasonable agreement. Experiments verify the existence of plastic hinge response for both internal stationary and moving hinges.

The analysis is limited to blast pressures between static collapse loads and pressures where the concrete begins to fracture and separate from the reinforcing elements. Additional study into loadings causing localized failure is recommended.

TABLE OF CONTENTS

TABLE OF CONTENTS, Concluded

LIST OF FIGURES

LIST OF FIGURES, Continued

LIST OF FIGURES, Concluded

LIST OF TABLES

LIST OF SYMBOLS

a	beam half span, plate short side in, (m)
$\overline{AR}$	aspect ratio for plates b/a, dimensionless
b	beam width, plate long side in, (m)
d	distance from tensile reinforcing element to opposite face in compression of cross section in bending in, (m)
F	multiplier on hinge moment designating fixed or simply supported beam ends or plate edges, F = 1 for simple supports, F = 2 for fixed supports Dimensionless
I	Beam moment of inertia lbf-in-sec^2(kg-m^2)
ℓ	plastic hinge width or deformed length in, (m)
m	mass per unit area of plate or beam lbf-sec^2/in^3, (kg/m^2)
M_o	plastic hinge moment, lbf-in(N-m)
M_u	plastic hinge moment per unit length, lbf(N)
P_E	uniform pressure over plate or beam lbf/in^2, (N/m^2,P_a)
P_C	maximum pressure of linear or nonlinear loading lbf/in^2, (N/m^2,P_a)
PE	potential energy lbf-in,(N - m)
q	reinforcement ratio, ratio of area of tensile re-inforcement per unit cross section area
R	radius of curvature of deformed hinge, in, (m)
S	surface area of plate or beam in^2(m^2)
t	time sec

w	weight per unit area, $lbf/in^2(N/m^2)$
W	work term lbf-in(N-m)
W_{int}	internal work lbf-in(N-m)
W_{ext}	External work lbf-in(N-m)
WF	work done by applied pressure lbf-in(N-m)
WP	work done by plastic hinge lbf-in,(N-m)
x	beam coordinate in(m)
x,y	plate coordinates in(m)
x_h	hinge length, in(m)
X	ratio x_h/a, dimensionless
Y	initial value of X, dimensionless
z	ratio of final hinge position of plate to b, dimensionless
α	time decay constant of pressure, dimensionless
β	spatial decay constant of pressure, dimensionless
δ	midpoint displacement for beams or plates, in(m)
Δ	change in hinge length due to rotation, in(m)
ε	strain, unit deformation, dimensionless
ε_b	strain due to bending, dimensionless
ε_t	strain due to tensile elongation, dimensionless
ε_u	ultimate material strain, dimensionless
n	sign of potential energy term (+) for explosive below horizontal structure, (-) for explosive above horizontal structure, (0) for a vertical structure
σ	stress, $lbf/in^2(N/m^2,P_a)$

σ_c concrete compressive strength, $lbf/in^2 (N/m^2, P_a)$

σ_r ultimate tensile strength reinforcing elements, $lbf/in^2 (N/m^2, P_a)$

θ rotation at hinge line, rad

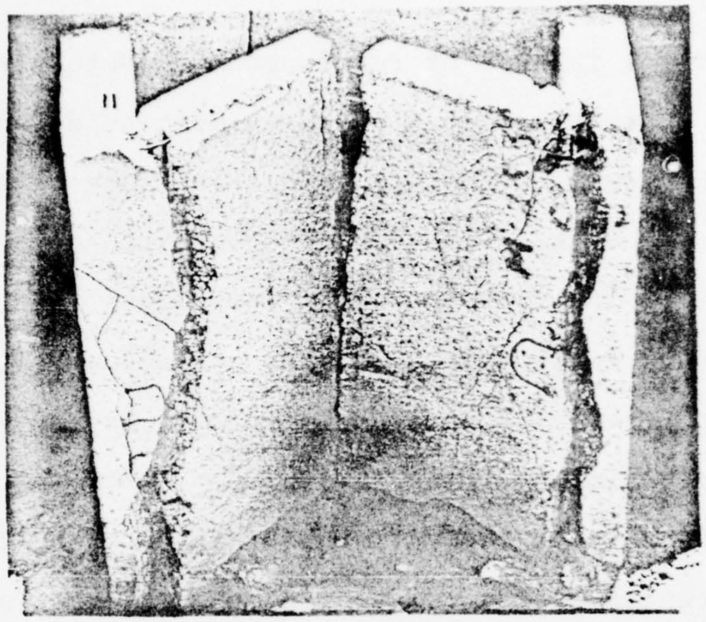
θ_u rotation corresponding to ultimate strain, rad

τ length of pressure time curve, sec

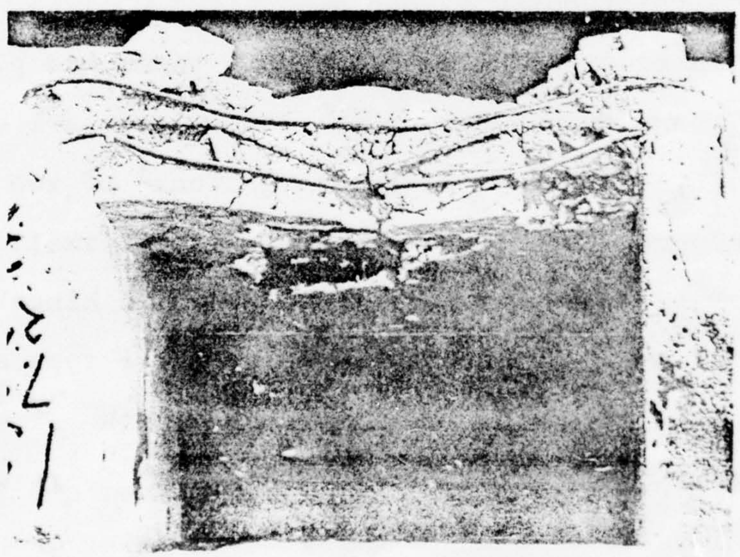
I. INTRODUCTION

Structural failure of reinforced concrete structures, as well as other kinds of structures must be defined in terms of the degree of damage. In this study failure is defined as the actual fracture or separation of a reinforcing element in the slab or beam. This means that before failure occurs considerable concrete cracking will have occurred and the reinforcing element will have yielded and reached the ultimate stress or strain. Since the tensile strength of concrete is negligible in comparison to the tensile strength of the reinforcing element the failure criterion is based solely on the ultimate stress or elongation of the reinforcing elements.

In this study reinforced concrete beams and plates subjected to transverse blast or impulsive loads are considered for analysis. Experimental investigations[1] of two edges fixed reinforced concrete slabs subjected to small underground near field explosions showed "plastic hinge" mechanisms at failure. Typical failures of these type are shown in Figure 1. Failure of this type suggest the same type failure mechanism as that observed in metal beams when subjected to impulsive loads of magnitude greater than the static collapse load of the member. For plates and beams



a) 36"x36"x4" two edges fixed
8.0 lbs @ 3 ft.



b) 24"x36"x4" two edges fixed
8.0 lbs @ 2 ft.

Figure 1. Typical failures of Reference 1.

with transverse or normal loadings where bending is predominant an idealized rigid perfectly plastic constitutive relation can be used to estimate the true moment carrying capacity of the structure. This assumption appears justified for the ductile steel reinforcing elements whose stress-strain curves show a rather small elastic portion when compared to the overall stress-strain curve for the material. The "fully plastic" or "ultimate moment" hinge as applied to statically loaded reinforced concrete structures is summarized by Szilard[2]. The "plastic hinge" or "fully plastic moment" is covered in detail by Timoshenko and Gere[3]. Abrahamson, et al,[4] used both stationary and moving "plastic hinges" to predict response of thin metal beams and plates when subjected to spatially uniform loads of several different time variations. Jones[5] used this simple method to successfully predict the large inelastic behavior of thin metal beams when subjected to impulsive loadings and found very good correlation with the experimental results of Menkes and Opat[6]. Both Florence[7] and the authors[8] have applied this method to reinforced concrete beams and reasonable agreement was found when analytical results of Reference 8 were compared to the experimental results of Reference 1. Based on the results obtained by the authors[8]

for beams with linear spatial loadings the study was extended to cover both linear and nonlinear impulsive spatial loadings for fixed and simply supported reinforced concrete beams and plates.

II. ANALYSIS

2.1 Assumed Response Mechanisms

The dynamic response of reinforced concrete plates and beams may well be broken down into four failure mechanisms as described below. In each particular case the failure mechanism exhibited by the structure is dependent on the initial magnitude of the assumed loading or pressure function. It is assumed that no response will occur unless the initial value of the loading pulse has exceeded the static collapse load of the structure based on a rigid perfectly plastic constitutive relation. The static collapse load is that load which will cause initial response of the plate or beam in the mechanism 1 mode described below.

Response under dynamic loadings may be characterized by the following four mechanisms.

Mechanism 1. This response, as shown in Figure 2, occurs when the initial magnitude of the loading pulse is slightly larger than the static collapse load for the member.

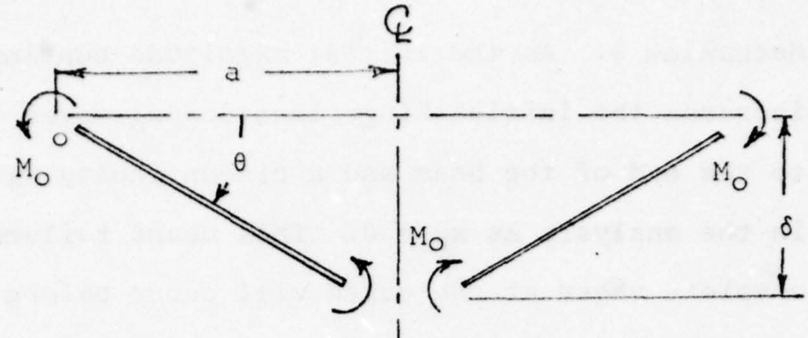


Figure 2. Failure by a mechanism 1 mode.

Mechanism 2. As the initial magnitude of the loading pulse increases a plastic hinge at interior points other than at the center of the member will form and move toward the center of the member. This mechanism type for a beam is shown in Figure 3.

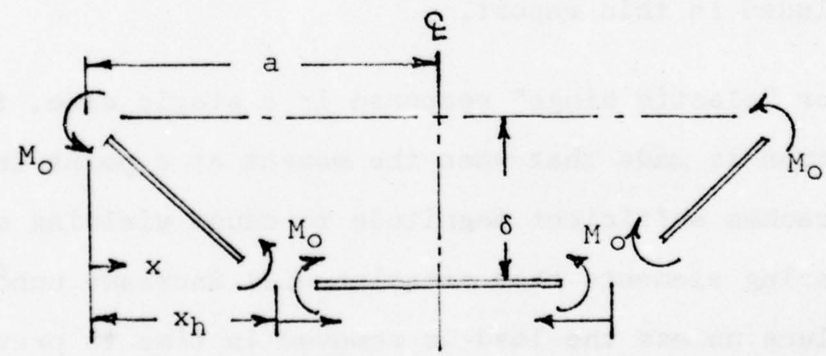


Figure 3. Mechanism 2 type beam response.

Mechanism 3. As the initial magnitude continues to increase the initial hinge location x_h moves closer to the end of the beam and a discontinuity will exist in the analysis as $x_h = 0$. This means failure by complete shear at the edges will occur before any hinge motion can occur. This type failure for metal beams has been discussed by Jones[5], however a general discussion of failure of reinforced concrete beams by this mechanism is not included in this report.

Mechanism 4. For very local blast loads on long beams or large plates, failure of the concrete begins before any appreciable overall response can occur and the concrete and reinforcing elements fail locally. This type mechanism is very similar to that of mechanism 3 above and discussion of this mechanism is also not included in this report.

For "plastic hinge" response in a static case, the assumption is made that when the moment at a point in the beam reaches sufficient magnitude to cause yielding of the reinforcing elements then rotation will increase unbounded to failure unless the load is removed in time to prevent total failure. This type failure mechanism may be described in terms of a uniform static load on a beam as shown in Figure 4.

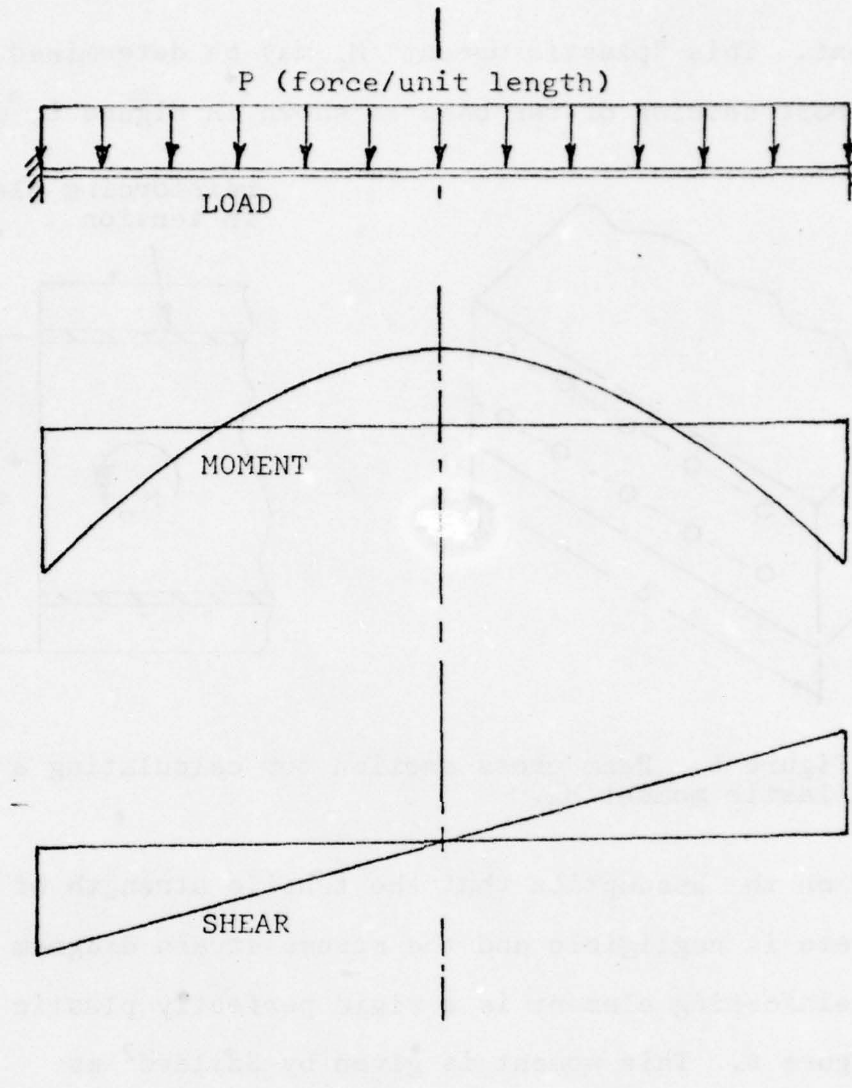


Figure 4. Load, shear and moment diagram for uniformly loaded (static) fixed ended beam.

The maximum positive bending moment occurs at the midspan and the maximum negative bending moment occurs at the ends of the beam. Rotation at the "hinge" begins when this moment is sufficient to cause yielding in the reinforcing

element. This "plastic moment" M_o may be determined from the cross section of the beam as shown in Figure 5, and is

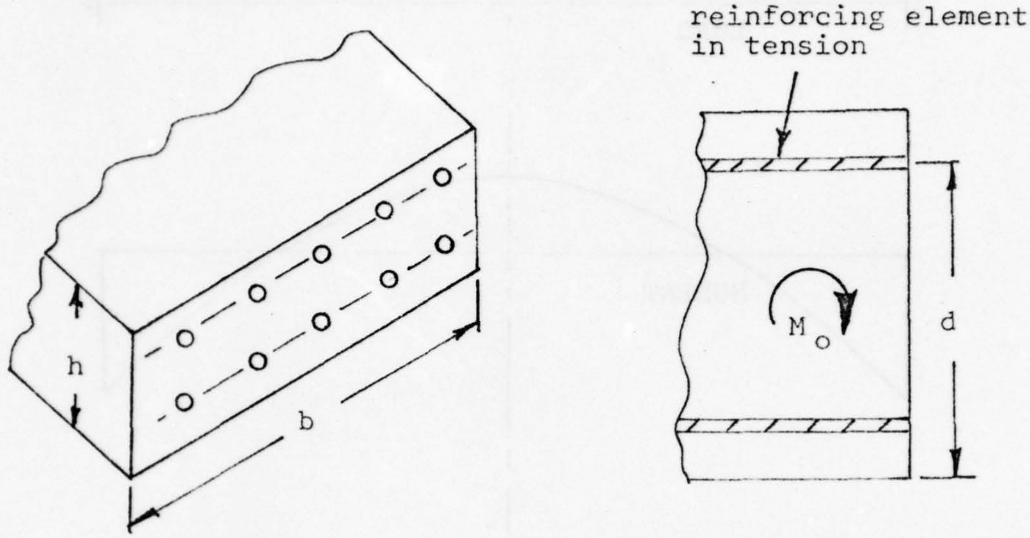


Figure 5. Beam cross section for calculating a plastic moment M_o.

based on the assumption that the tensile strength of the concrete is negligible and the stress-strain diagram of the reinforcing element is a rigid perfectly plastic type of Figure 6. This moment is given by Szilard[2] as

$$M_o = 0.9[d^2 bq\sigma_r(1-0.59q\sigma_r/\sigma_c)] \; . \tag{1}$$

The hinge moment per unit length is defined as

$$M_u = M_o/b \; , \tag{2}$$

where d and b are shown in Figure 5, σ_r is the ultimate

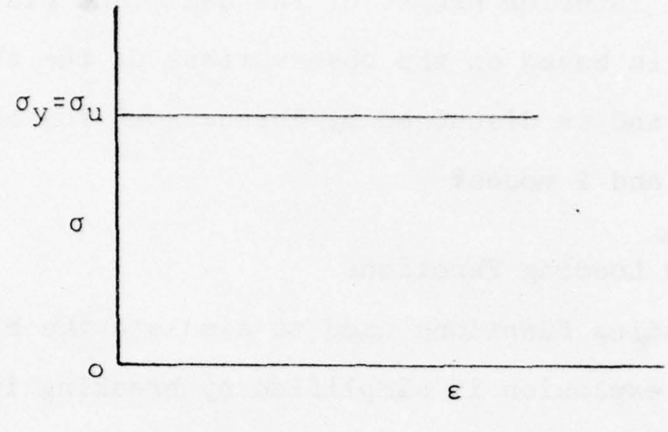


Figure 6. Assumed stress strain curve of the reinforcing elements.

stress of the reinforcing element, σ_c, is the compressive strength of the concrete, and q is the reinforcing ratio, i.e., the area of reinforcing elements in tension to the total cross sectional area of the beam.

When the moment at the midspan reaches M_o, localized rotation occurs at the ends and midspan and the resulting response is as shown in Figure 2. For a static case rotation will occur until fracture of the reinforcing element occurs. Rotation for failure of the reinforcing elements will be discussed later. The remainder of this report is restricted to discussion of mechanisms 1 and 2 dynamic response in beams and plates when subjected to spatially and time varying loading functions. One very important and underlying assumption used in this analysis is that the

shear at all interior hinges of the beams and plates is zero. This is based on the observations of the shear diagram of Figure 4 and is discussed by Abrahamson[4] for both the mechanism 1 and 2 modes.

2.2 Assumed Loading Functions

The loading functions used to simulate the blast or underground explosion is simplified by breaking it down into two parts with separable time and spatial variables. The general expression for the loading is given as

$$P(x,y,t) = f(t)P(x,y) \tag{3}$$

where x,y are the spatial coordinates and t is the time.

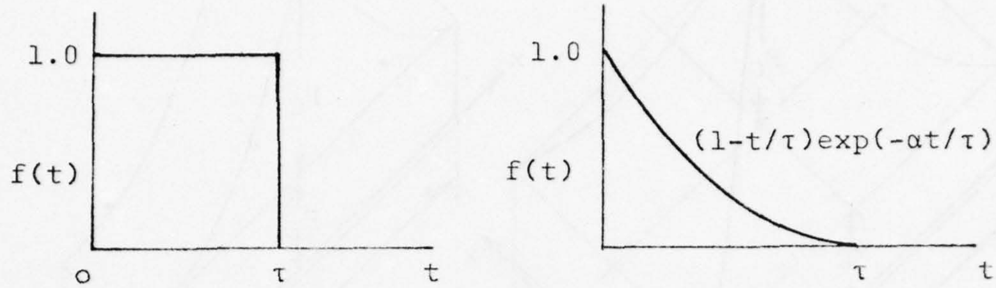
Two types of time variations were chosen for the analysis. The fundamental square wave or impulse of the form

$$\begin{aligned} f(t) &= 1 \quad , \quad 0 \leq t \leq \tau \\ f(t) &= 0 \quad , \quad t > \tau \end{aligned} \tag{4}$$

and a decay type

$$\begin{aligned} f(t) &= (1-t/\tau)\exp(-\alpha t/\tau) \quad 0 \leq t \leq \tau \\ f(t) &= 0 \quad , \quad t > \tau \end{aligned} \tag{5}$$

where τ is the pulse duration in time and α is the time decay constant. These two forms are shown schematically



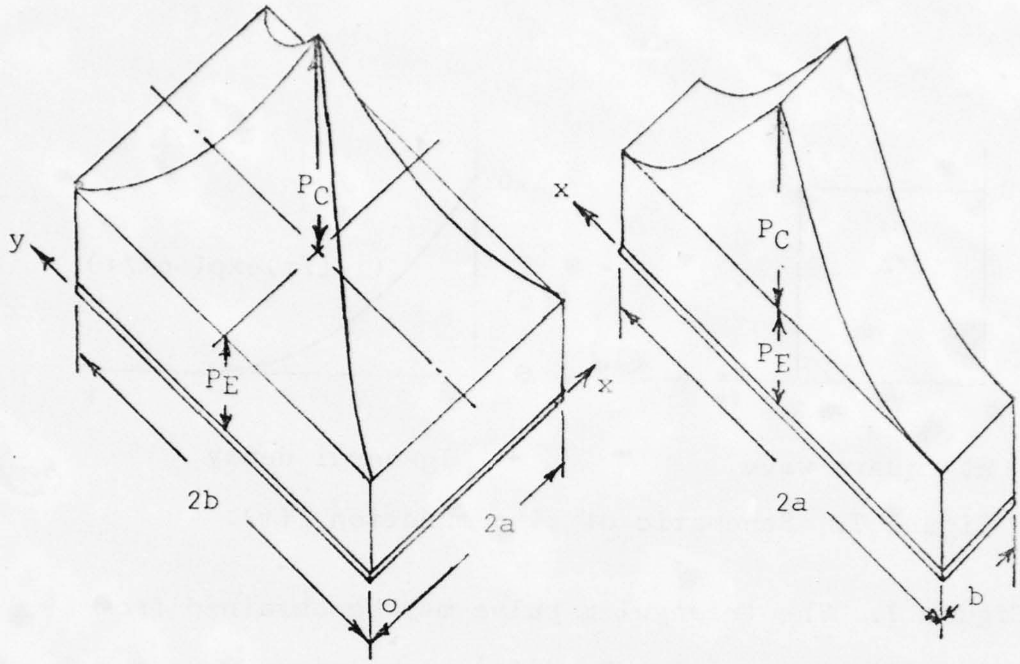
a) square wave   b) general decay

Figure 7. Schematic of time function f(t).

in Figure 7. The triangular pulse may be obtained from Equation (5) when $\alpha = 0$. No difficulty arises when $\alpha \to 0$ as long as the time integrals are accomplished numerically, however if a general closed form solution is required then the triangular or linear decay time function must be handled as a special case. This problem arises due to the $1/\alpha$ appearing in the result of integrating Equation (5) with respect to time.

In this study the spatial integration was handled in closed form and the spatial loading function P(x,y) was assumed to be of the symmetric form of Figure 8 and given as

$$\begin{aligned} P(x) &= P_E + P_C \frac{x}{a} \exp\beta\left(\frac{x}{a}-1\right), \quad 0 \le x \le a \\ P(y) &= P_E + P_C \frac{y}{b} \exp\beta\left(\frac{y}{b}-1\right), \quad 0 \le y \le b \end{aligned} \tag{6}$$



a) plate b) beam

Figure 8. Schematic of spatial loadings.

for the plates and

$$P(x) = P_E + P_C \frac{x}{a} \exp\beta\left(\frac{x}{a}-1\right) , \quad 0 \le x \le a \tag{7}$$

for beams where P_E is a uniform load acting over the whole structure, P_C is the magnitude at the center for the non-linear superimposed pressure and β is the spatial decay constant. Schematics, showing the coordinates (x,y) and dimensions a,b, of both plate and beam spatial loadings are given in Figure 8. The linearly varying loads are special cases of Equations (5) and (6) where $\beta = 0$. The spatial

integration was handled in closed form and a separate case for $\beta = 0$ was included in the analysis. In Equation (5) the same decay constant β was used for both the x and y variations. This would be true only for a central symmetric load on a square plate. However, due to uncertainities in both the mechanics model and loading functions the assumption appears reasonable for aspect ratios of at least two or less. Both the spatial and time decay terms α and β are obtained from observations of either experimental or analytically determined pressure time histories of under ground blasts.

2.3 Equations of Motion

2.3.1 Introduction

The equations of motion for both beams and plates are based on a conservation of energy. The equations as presented here were derived from force balance equations and verified by comparing with independent calculations on individual energy terms. Beams whose response starts with interior hinges not at the center (mechanism 2) are assumed to respond in this manner until the moving hinges reach the beam center and respond in a mechanism 1 mode until the rotation stops. For plates the location for the mechanism 1 response is dependent on both the load shape and plate aspect ratio and may be determined by a method described by Szilard[2]. For

plates starting in a mechanism 2 response, it is assumed that the final hinge locations will be that determined as if the plate assumed a mechanism 1 mode. This means that for each plate a final hinge location must be calculated for each loading. The method for doing this is discussed along with the derivation of the equations of motion for the plate.

2.3.2 Beams

Since the beam response is either mechanism 1 or mechanism 2 with interior hinges moving toward the center, the derivation will be based on a mechanism 2 mode and the mechanism 1 response will be a special case when the hinges start at the midspan of the beam. The beam loading and response is assumed to be completely symmetrical therefore only half the beam need be considered. Using the response shape of Figure 9 the derivation of the beam equations of motion is as follows. For the exterior portion of length x_h the general equation is

$$I\ddot{\theta} = \int P(x,t)x\,dS - n\int wx\,dS - FM_o \tag{8}$$

where I is the moment of inertia and $\ddot{\theta}$ the angular acceleration about the axis at 0, w is the weight per unit area of surface area, S is the surface area over which the pressure is acting, n is a position indicator with n=0 for a vertical wall, n=-1 for blast above beam and n=+1 for blast below beam,

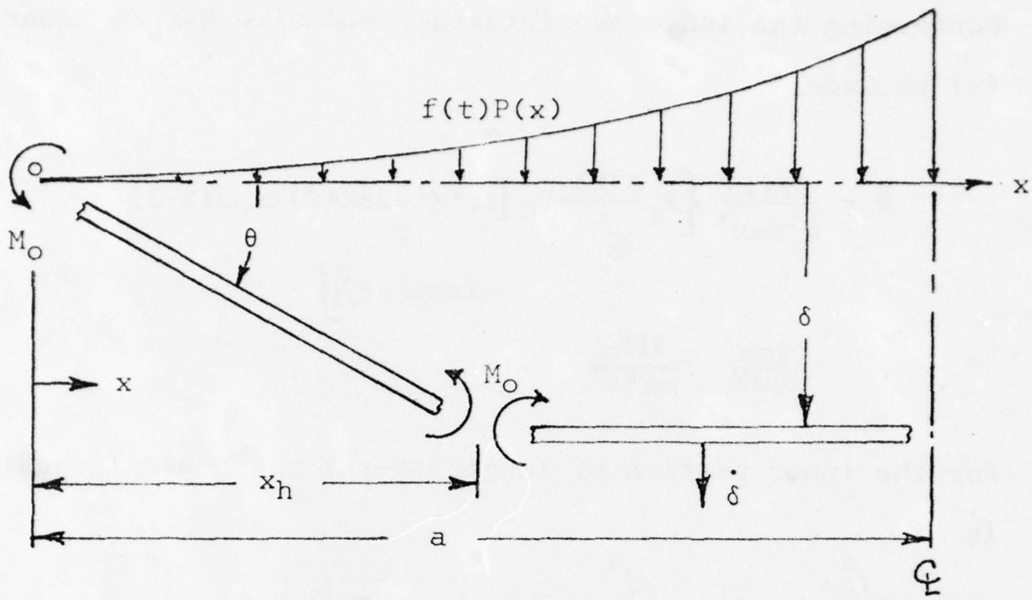


Figure 9. Schematic for beam response of mechanism 2.

and $\bar{F}$ is an end fixity term with F=2 for fixed ended beams and F=1 for simply supported beams. Using Equations (2) and (7) with m the mass per unit of surface area, w the weight per unit of surface area and dS=b dx Equation (8) becomes

$$\ddot{\theta}\int_0^{x_h} bmx^2\,dx = f(t)\int_0^{x_h} b\left[P_E+P_C\left(\frac{x}{a}\right)\exp\beta\left(\frac{x}{a}-1\right)\right]x\,dx$$

$$-n\int_0^{x_h} bwx\,dx \;-b\bar{F}M_u\,. \tag{9}$$

Performing the indicated integrals and with $X = x_h/a$ Equation (9) becomes

$$\ddot{\theta} = \frac{3f(t)}{\beta^3 m a x^3}\left\{P_E\frac{\beta^3 X^2}{2} + P_C\left[(\beta^2X^2 - 2\beta X + 2)\exp\beta(X-1) - 2\exp(-\beta)\right]\right\} - \frac{3nw}{2maX} - \frac{3FM_u}{ma^3X^3} \quad . \tag{10}$$

For the inner portion of length $(a-x_h)$ the general equation is

$$\ddot{\delta}\int_{x_h}^{a} bm\,dx = f(t)\int_{x_h}^{a} b\left[P_E + P_C\left(\frac{x}{a}\right)\exp\beta\left(\frac{x}{a}-1\right)\right]dx - n\int_{x_h}^{a} bw\,dx \tag{11}$$

Performing the indicated integrations Equation (11) becomes

$$\ddot{\delta} = \frac{f(t)}{m\beta^2}\left\{P_E\beta^2 + \frac{P_C}{1-X}\left[\beta - 1 - (\beta X - 1)\exp\beta(X-1)\right]\right\} - \frac{nw}{m} \tag{12}$$

In order to maintain continuity between the two elements

$$\begin{gathered} \theta x_h = \theta a X = \delta \\ \text{or} \\ \dot{\theta} x_h = \dot{\theta} a X = \dot{\delta} \quad . \end{gathered} \tag{13}$$

The assumption is made that Equations (10), (12) and (13) satisfy all the conditions required for motion by plastic

hinges and in order to determine the initial response the initial conditions

$$\left.\begin{aligned}\ddot{\theta}aX &= \ddot{\delta}\\ f(t) &= 1\end{aligned}\right\} t = 0 \tag{14}$$

are imposed on equations (10) and (12). The resulting equation becomes

$$a^2\beta^3P_EY^2 - nw\beta^3a^2Y^2 - 6FM_u\beta^3 \tag{15}$$

$$+ 2a^2P_C\left\{\frac{1}{1-Y}[1-\beta+(\beta Y-1)\exp\beta(Y-1)]+3[(\beta^2Y^2-2\beta Y+2)\exp\beta(Y-1) -2\exp(-\beta)]\right\} = 0 .$$

Where Y, the initial value of $X(x_{ho}/a)$ determines the mechanism for initial response. Only solutions for $0<Y\leq 1$ are allowed and solutions where $Y>1$ are set $Y=1$. The solution $Y=1$ means a mechanism 1 response and for this case x_h is set equal to a in Equation (10) and the response is determined from this equation only and Equation (12) is disregarded. For the case of a linear spatial variation for the beam, i.e., if $\beta=0$ in Equation (7), then the equation of motion must be derived for this case and the $\ddot{\theta},\ddot{\delta}$ equations similar to Equations (10) and (12) are found to be

$$\ddot{\theta} = \frac{f(t)}{2maX}(3P_E+2XP_C) - \frac{3nw}{2maX} - \frac{3FM_u}{ma^3X^3} \tag{16a}$$

$$\ddot{\delta} = \frac{f(t)}{m}\left[P_E + \frac{P_C}{2}(X+1)\right] - \frac{nw}{m} \quad (16b)$$

Applying the same initial conditions as in Equation (14) and the same procedure for finding the initial response of Equations (10) and (12) then the beam response for $\beta=0$ may be determined.

Assuming conservation of energy the equations of motion can be double checked using the relation

$$\dot{KE} = \dot{WF} - \dot{WP} - \dot{PE} \quad (17)$$

where the ($\cdot$) means time derivative, KE is kinetic energy, WF work done by applied pressure or force, WP work done by plastic hinge and PE is potential energy. In that it is desirous to monitor these values for given response calculations, terms for the right hand side of Equation (17) were derived. For the nonlinear spatial function ($\beta\neq0$) the terms are

$$\dot{WF} = \frac{2ba^2\dot{\theta}f(t)}{\beta^3}\left\{P_E\beta^3X(1-\frac{X}{2}) + P_C[(2-\beta X)\exp\beta(X-1)-2\exp(-\beta) + \beta X(\beta-1)]\right\}$$

$$\dot{WP} = 2bFM_u\dot{\theta} \quad (18)$$

$$\dot{PE} = 2a^2bnwX(1-\frac{X}{2})\dot{\theta}$$

where w is the weight per unit area. The work and energy

terms for the linear spatial loading (β=0) are given as

$$\dot{WF} = 2a^2bf(t)\dot{\theta}\left[P_E X\left(1-\frac{X}{2}\right) + \frac{P_C}{2}X\left(1-\frac{X^2}{3}\right)\right]$$

$$\dot{WP} = 2bFM_u\dot{\theta} \tag{19}$$

$$\dot{PE} = 2a^2bnwX\left(1-\frac{X}{2}\right)\dot{\theta}$$

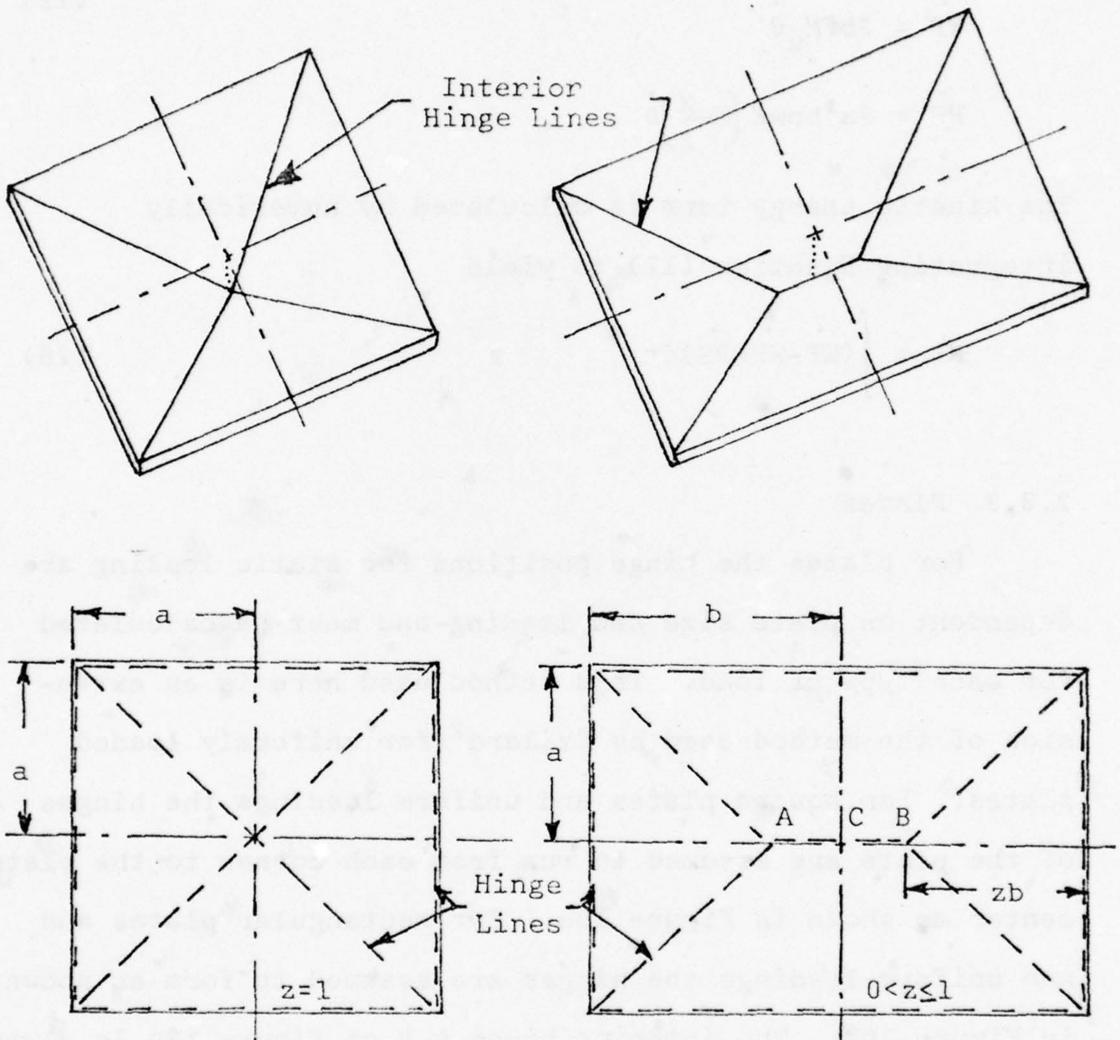
The kinetic energy term is calculated by numerically integrating Equation (17) to yield

$$KE = \int(\dot{WF}-\dot{WP}-\dot{PE})dt \tag{20}$$

2.3.3 Plates

For plates the hinge positions for static loading are dependent on plate size and loading and must be calculated for each type of load. This method used here is an extension of the method used by Szilard[2] for uniformly loaded plates. For square plates and uniform loadings the hinges of the plate are assumed to run from each corner to the plate center as shown in Figure 10a. For rectangular plates and non uniform loadings the hinges are assumed to form as shown in Figure 10b. The interior hinge A-B of Figure 10b is always assumed to run parallel to the long side of the plate. Schematics of the deformed shape are shown along with the

plan views of Figure 10. The position zb of the interior hinge was determined using the minimum energy for the assumed shape of Figure 10b. This method is outlined below using a



a) square plate b) rectangular plate

Figure 10. Hinge location for static mechanism 1 mode for plates.

quarter of the plate shown in Figure 11. Justification of this assumption is based on a symmetric response shape and loading

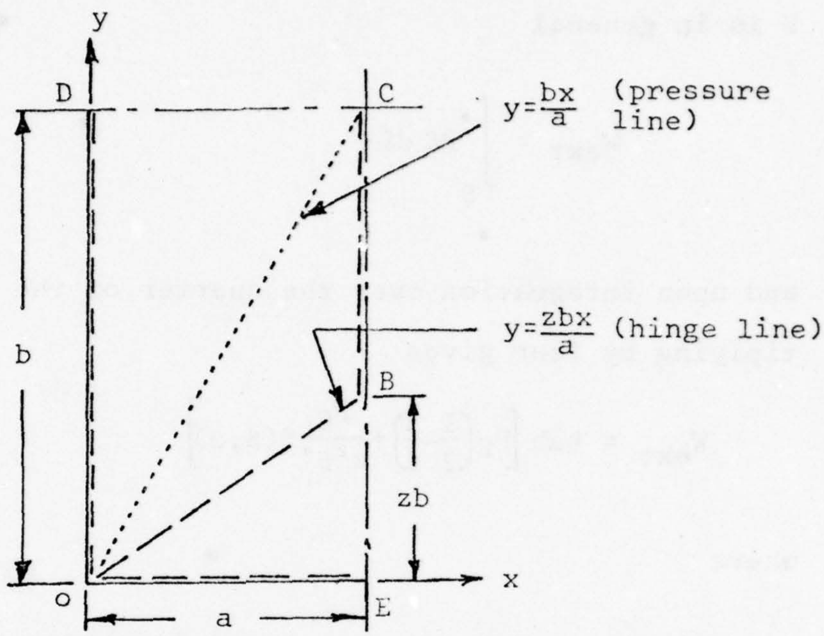


Figure 11. Schematic used for calculating final hinge locations.

model. Equating the external work to the internal work for a given unit center displacement yields an equation in terms of the plate dimensions, loading parameters, and the internal hinge location zb. For the derivation, θ is assumed to be the rotation along the line OD and the assumption of constant displacement along BC the rotation about line OE is $\theta a/zb$. For the energy approach the displacement along BC is arbitrary and chosen as $\delta = 1$ for simplification. The displacement along the line EB is then $\delta = y/zb$ and the

displacement along the line DC is $\delta = x/a$. The external work done by the pressure load of Equation (6) over the area S is in general

$$W_{ext} = \int_S P\delta \, dS \tag{21}$$

and upon integration over the quarter of the plate and multiplying by four gives

$$W_{ext} = 4ab\left[P_E\left(\frac{1}{2}-\frac{z}{6}\right)+\frac{P_C}{z^2\beta^4}f(\beta,z)\right] \tag{22}$$

where

$$f(\beta,z) = (3-\beta z)\exp\beta(z-1)-(2\beta z^2+9z^2+2\beta z+3)\exp(-\beta) + 2\beta^2-7\beta z^2+9z^2 \quad . \tag{23}$$

The internal work is the work done by the plastic hinge moving through some local angle at the hinge lines. The work done by the hinge lines that move diagonally are based on the projected component on the coordinate axes. For instance the line OB of Figure 11 does work an amount of $(M_u a)(\theta a/zb)$ about an axis parallel to ox and an amount of $(M_u zb)(\theta)$ about an axis parallel to the oy axis. Based on this assumption and depending on the edge conditions the internal work may be expressed as

$$W_{int} = 4FM_u\left(\frac{a}{zb}+\frac{b}{a}\right) \tag{24}$$

where $\theta a = 1$ for the assumed displacement $\Delta = 1$. Equating W_{ext} to W_{int} and rearranging terms yields

$$\frac{P_C b^2}{FM_u} = \frac{(z+\overline{AR}^2 z^2)\beta^4}{f(\beta,z)+K\beta^4 z^2\left(\frac{1}{2}-\frac{z}{6}\right)}, \quad K = \frac{P_E}{P_C}. \tag{25}$$

If the left hand side of Equation (25) is plotted against z a minimum value is realized. It was found analytically that only one minimum existed for $0\leq z\leq 1$ therefore the value of z for a minimum can be determined by differentiating Equation (25) with respect to z and setting the right side to zero. The results of this for the nonlinear spatial load is

$$\overline{AR}^2 z\left\{P_E\beta^4 z^2\left(\frac{z}{6}\right)+P_C[(6-4\beta z+\beta^2 z^2)\exp\beta(x-1)-(6+2\beta z)\exp(-\beta)]\right\}$$
$$+P_E\beta^4 z^2\left(\frac{z}{3}-\frac{1}{2}\right)+P_C[(3-3\beta z+\beta^2 z^2)\exp\beta(z-1)+(2\beta z^2+9z^2-3)\exp(-\beta)]$$
$$-2\beta^2 z^2+7\beta z^2-9z^2 = 0 , \tag{26}$$

and for the linear spatial load

$$P_E(4\overline{AR}^2 z+8z-12)+P_C(2\overline{AR}^2 z^3+3z^2-5) = 0 . \tag{27}$$

The solution of Equations (26) or (27) for z determines the hinge locations for a mechanism 1 response and final locations for the mechanism 2 response in plates.

The schematic of a mechanism 1 plate response would be identical to that of Figures 10 and 11 and the schematic of a mechanism 2 plate response is shown in Figure 12.

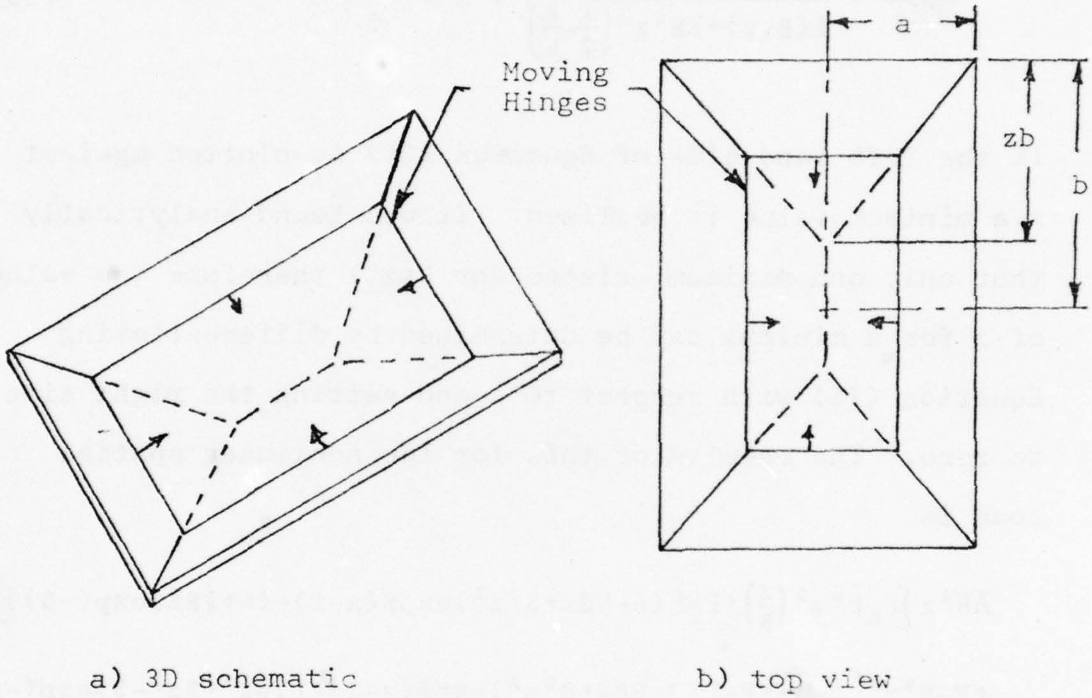


Figure 12. Mechanism 2 plate response.

As in the case for the beams, only the derivation of the mechanism 2 response will be given and when x_h = a only the equation of motion for $\ddot{\theta}$ will be used. For this derivation many of the same terms used for the beam will also be

used here. Most of these terms are shown in Figure 13 and only a quarter of the plate is shown due to symmetry of loading and response. The assumption for mechanism 2 is

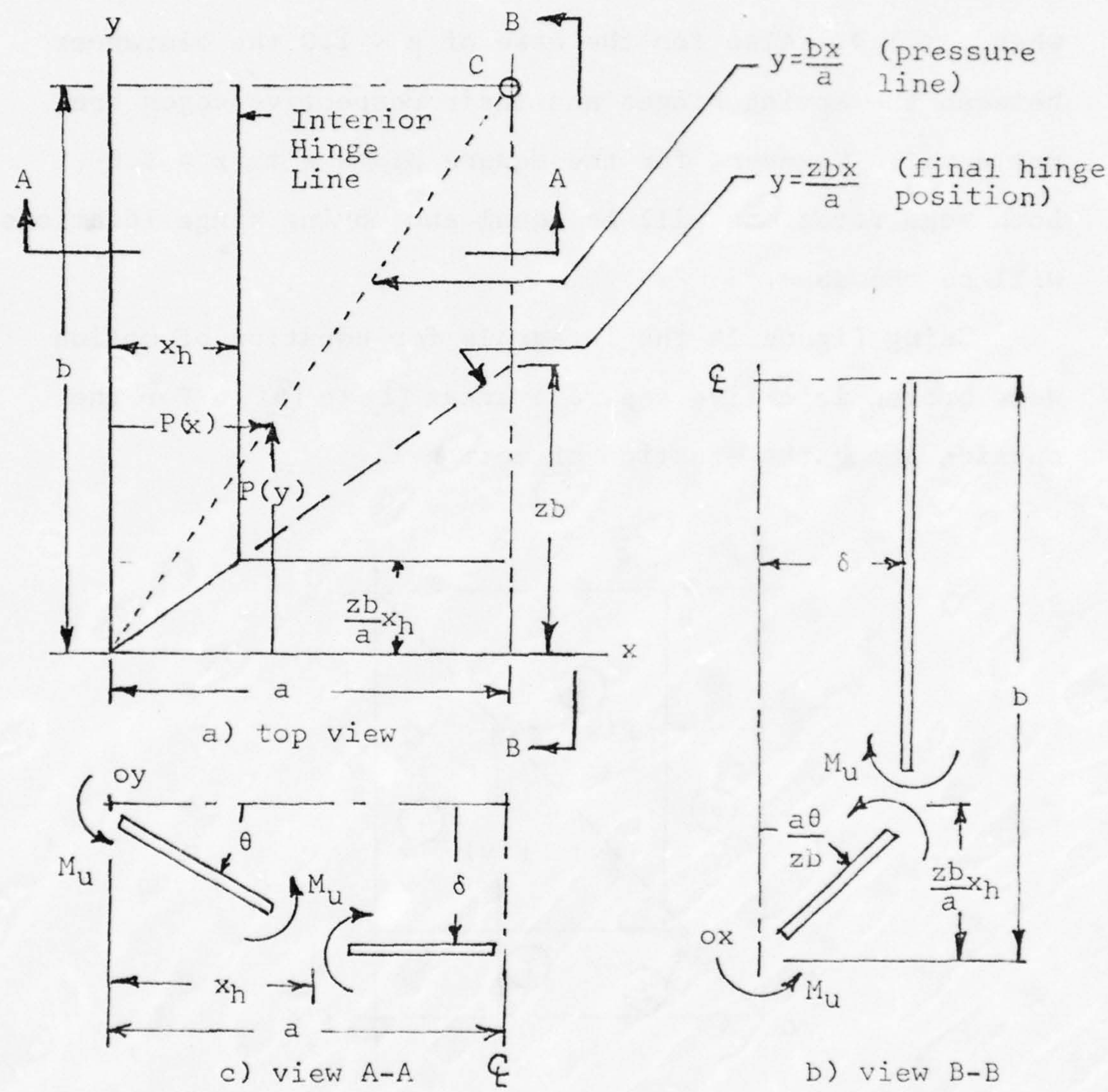


Figure 13. Schematic for derivation of equations of motion of plate.

that the inner portion of the plate is flat and has a displacement δ, same as that of the plate center point. This insures continuity of the inner hinge line and as shown in Figure 13a gives different rotations for the plate edges when $z < 1.0$. Also for the case of $z < 1.0$ the distances between the moving hinges and their respective edges are not equal. However, for the square plate with $z = 1.0$ both edge rotations will be equal and moving hinge locations will be the same.

Using Figure 14 the integrals for equation of motion were broken into five separate areas (1) to (5) . For the outside areas the equation of motion is

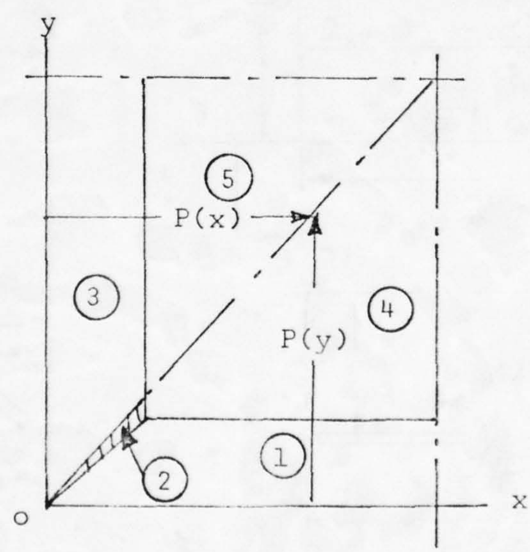


Figure 14. Regions used in integrals of equation of motion.

$$
\begin{aligned}
1\;&\left\{
\begin{aligned}
&\int_0^{\frac{zb}{a}x_h} m\left(\frac{a}{zb}\ddot{\theta}\right)\left(a-\frac{ya}{zb}\right)y^2\,dy \;-f(t)\int_0^{\frac{zb}{a}x_h}\left\{P_E+P_C\left(\frac{y}{b}\right)\left[\exp\beta\left(1-\frac{y}{b}\right)\right]\left(a-\frac{a}{zb}\right)\right\}y\,dy\\
&+\int_0^{\frac{zbx_h}{a}} nw\left(a-\frac{ya}{zb}\right)y\,dy \;+FM_u a
\end{aligned}\right.\\
2\;&\left\{
\begin{aligned}
&+\int_0^{x_h} m\ddot{\theta}\left(\frac{bx}{a}-\frac{zbx}{a}\right)x^2\,dx \;-f(t)\int_0^{x_h}\int_{\frac{zbx}{a}}^{\frac{bx}{a}}\left\{P_E+P_C\left(\frac{y}{b}\right)\left[\exp\beta\left(1-\frac{y}{b}\right)\right]\right\}x\,dy\,dx\\
&+\int_0^{x_h} nw\left(\frac{bx}{a}-\frac{zbx}{a}\right)x\,dx
\end{aligned}\right.\\
3\;&\left\{
\begin{aligned}
&+\int_0^{x_h} m\ddot{\theta}\left(b-\frac{bx}{a}\right)x^2dx \;-f(t)\int_0^{x_h}\left\{P_E+P_C\left(\frac{x}{a}\right)\left[\exp\beta\left(1-\frac{x}{a}\right)\right]\left(b-\frac{bx}{a}\right)\right\}x\,dx\\
&+\int_0^{x_h} nw\left(b-\frac{bx}{a}\right)dx \;+\;FM_u b \qquad\qquad = \;0
\end{aligned}\right.
\end{aligned}
\tag{28}
$$

Using $\overline{AR} = b/a$, $X = x_h/a$, evaluating the integrals and rearranging terms Equation (28) becomes

$$m\ddot{\theta}a^4\overline{AR}X^3\left[(z^2\overline{AR}+1)\left(\frac{1}{3}-\frac{X}{4}\right)+(1-z)\frac{X}{4}\right]$$

$$= \frac{f(t)a^3\overline{AR}}{z^2\beta^4}\Big\{P_E\beta^4z^2X^2\left[\overline{AR}z^2\left(\frac{1}{2}-\frac{X}{3}\right)+\frac{1}{3}(1-z)X+1\right]$$

$$+P_C[-(2\overline{AR}\beta z^2+2\beta z^2+9z^2+6\overline{AR}z-3)\exp(-\beta)$$

$$+(\beta^3z^2X^2-\beta^3z^2X^3+4\beta^2z^2X^2)\exp\beta(X-1)$$

$$+(-2\beta^2z^2X-9\beta z^2X+2\beta z^2+9z^2)\exp\beta(X-1) \tag{29}$$

$$+\overline{AR}(-\beta^3z^4X^3+\beta^3z^4X^2+3\beta^2z^3X^2$$

$$-2\beta^2z^3X-6\beta z^2X+2\beta z^2+6z)\exp\beta(zX-1)$$

$$+(-\beta^2z^2X^2+3\beta zX-3)\exp\beta(zX-1)]\Big\}$$

$$-nwa^3\overline{AR}X^2\left[\overline{AR}z^2\left(\frac{1}{2}-\frac{X}{3}\right)+ (1-z)\frac{X}{3}+\frac{1}{2}-\frac{X}{3}\right]$$

$$- FM_u(1+\overline{AR})a \quad .$$

For the inner flat portion the integrals are broken into 2 parts, i.e., areas 4 and 5. In this case the interia term is much simpler and is included under the same integral as the potential energy term.

$$(m\ddot{\delta}+nw)(a-x_h)\left(b-\frac{zb}{a}x_h\right)$$

$$④ \quad = f(t)\int_{x_h}^{a}\int_{\frac{zb}{a}x_h}^{\frac{bx}{a}}\left\{P_E+P_C\left(\frac{y}{b}\right)\left[\exp\beta\left(\frac{y}{b}-1\right)\right]\right\}dy\,dx \tag{30}$$

$$⑤ \quad +f(t)\int_{\frac{b}{a}x_h}^{b}\int_{x_h}^{\frac{ay}{b}}\left\{P_E+P_C\left(\frac{x}{a}\right)\left[\exp\beta\left(\frac{x}{a}-1\right)\right]\right\}dy\,dx$$

Evaluating the integrals and rearranging terms Equation (30) becomes

$$\begin{aligned}\bar{m}(1-X)(1-zX)\ddot{\delta} = f(t)\Big\{&P_E(1-zX+zX^2-X)\\ &+\frac{P_C}{\beta^3}[2\beta-4+(\beta+\beta^2zX^2-\beta^2zX-\beta)\exp\beta(zX-1)\\ &+(4+\beta-3\beta X-\beta^2X+\beta^2X^2)\exp\beta(X-1)]\Big\}\\ &-nw(1-X)(1-zX)\quad .\end{aligned} \tag{31}$$

Using the same approach as in the case of the beams the kinetic energy, plastic work, potential energy and pressure work of the plate with nonlinear spatial loads ($\beta\neq0$) may be calculated using the following equations.

$$\dot{WF} = \frac{4\dot{\theta}f(t)\overline{AR}a^3}{Z^2\beta^4}\Bigg\{P_E\beta^4z^2\left(X-\frac{zX^2}{2}+\frac{zX^3}{3}\ \frac{X^3}{3}\right)$$

$$+P_C[2\beta^2z^2X-4\beta z^2X)$$

$$+(-\beta^2z^2X+\beta^2z^2X^2-3\beta zX+2\beta z+3)\exp\beta(zX-1)$$

$$+(-5\beta z^2X+\beta^2z^2X^2-\beta^2z^2X+2\beta z^2+9z^2)\exp\beta(X-1)$$

$$-(2\beta z^2+2\beta z+9z^2+3)\exp(-\beta)]\Bigg\} \tag{32}$$

$$\dot{PE} = 4nw\overline{AR}a^3\dot{\theta}\left(X-\frac{X^2}{2}-\frac{zX^2}{2}+\frac{zX^3}{3}\right)$$

$$\dot{WP} = 4F\dot{\theta}M_u a\left(\frac{\overline{AR}^2z+1}{z\overline{AR}}\right) \quad .$$

For the linear spatial loading (β=0) the same procedure as above was followed in deriving the equations of motion for the plate. Again, the reason for the separate derivation for β=0 is that in the nonlinear spatial loading equations such as Equations (29) and (31) the β term appears in the denominator. In these equations L'Hospital rule could be used as $\beta \to 0$, however it appeared to be much easier and straight forward to simply derive the equations from basic assumptions. With these comments in mind and using the same general procedure leading to Equations (29) and (31)

the equations of motion for a plate with $\beta=0$ are

$$m\ddot{\theta}a\,\overline{AR}X^3\left[(z^2\overline{AR}+1)\left(\frac{1}{3}-\frac{X}{4}\right)+(1-z)\frac{X}{4}\right]$$

$$= f(t)a^3\overline{AR}X^2\Big\{P_E\left[(z^2\overline{AR}+1)\left(\frac{1}{2}-\frac{X}{3}\right)+(1-X)\frac{X}{3}\right]$$

$$+P_C\Big[z^3X\overline{AR}\left(\frac{1}{3}-\frac{X}{4}\right)+\frac{X^2}{4}\left(\frac{11}{6}-\frac{z^2}{2}-X\right)\Big]\Big]\Big\} \tag{33}$$

$$-nwa^3\overline{AR}X^2\left[\overline{AR}z^2\left(\frac{1}{2}-\frac{X}{3}\right)+(1-z)\frac{X}{3}+\frac{1}{2}-\frac{X}{3}\right]$$

$$-FM_u(1+\overline{AR})a \quad ,$$

and

$$m(1-zX)\,\ddot{\delta} = f(t)\left\{P_E(1-zX)+P_C\left[\frac{1}{3}(X+1)-\frac{X}{2}\left(z^2+\frac{1}{3}\right)\right]\right\}$$

$$-nw(1-zX) \quad , \tag{34}$$

with continuity assured by $\dot{\theta}Xa = \dot{\delta}$. The corresponding equations for $\dot{WP}$ and $\dot{PE}$ are the same as Equation (32) except for $\dot{WF}$ which is

$$\dot{WF} = 4f(t)\dot{\theta}\overline{AR}a^3\Big[P_E\left(X-\frac{X^2}{2}-\frac{zX^2}{2}+\frac{zX^3}{3}\right)$$

$$+P_C\left(\frac{X}{3}-\frac{X^3}{6}-\frac{zX^3}{6}+\frac{X^4}{6}+\frac{z^2X^4}{4}-\frac{zX^4}{4}\right)\Big] \quad . \tag{35}$$

The general solution procedure here is the same as for beams. After the inner hinge position z has been determined by either Equation (26) or (27) it becomes a constant for the equations of motion. In order to determine the initial positions for the inner hinges the initial conditions

$$\left.\begin{aligned} \ddot{\theta} a X &= \ddot{\delta} \\ f(t) &= 1 \end{aligned}\right\} \quad t = 0 \tag{36}$$

are imposed on the Equations (29) and (31) or Equations (33) and (34). Due to the length and complexity of these equations they were omitted but included in the numerical calculations to be discussed later. The solution resulting from these initial conditions determines the initial mechanism to be used. Again assuming Y is the initial value of $x_h/a(x_{h_0}/a)$, then for $0<Y<1$ a mechanism 2 response is initiated and the response is governed by both the $\ddot{\theta}$ and $\ddot{\delta}$ equations until the inner hinges reach the final positions with $x_h=a$. Also for Y=1 the initial response is a mechanism 1 type and only Equations (29) or (33) would be applicable for a plate problem.

This concludes the derivation of the equations of motion necessary for determining the dynamic response under the assumed plastic hinge motion. The range of applicability of these equations and their shortcomings will be discussed later under a separate section but before that a section describing a proposed failure criterion will be given.

2.4. Failure Criterion

Under the assumptions used in the previous sections the rotations are assumed to be highly localized and mathematically the hinges are assumed to have zero width. In a real or true structure the hinge width or damaged area must be non zero due to the finite thickness of the structure. Intuitively, the thicker the structure the greater the hinge width due to the assumed bending taking place. For an assumed damaged or deformed width, at a hinge location, the total strain may be estimated.

The tensile strain in the reinforcing element at a hinge position may be broken into two parts, i.e., the strain due to bending ε_b and the strain due to axial elongation in tension ε_t. The total strain will then be the sum of these or

$$\varepsilon = \varepsilon_b + \varepsilon_t \quad . \tag{37}$$

For a given assumed damage or hinge width ℓ the strain terms may be approximated using the diagram of Figure 15. Using a known rotation θ and an arc length ℓ the radius of curvature R at the hinge becomes

$$R = \frac{\theta}{\ell} \tag{38}$$

and assuming strain due to bending as proportional to the

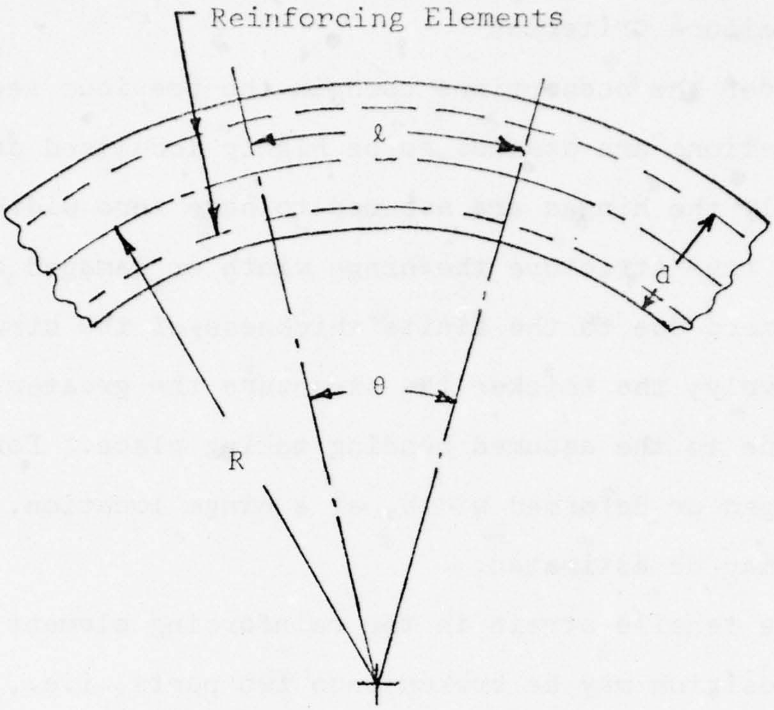


Figure 15. Schematic of plate or beam thickness showing rotation, placement of reinforcing elements and assumed hinge length.

distance from the assumed neutral axis then

$$\varepsilon_b = \frac{\theta d}{2\ell} \qquad (39)$$

where d is the distance between the tensile reinforcing element and outer compressive surface.

For a mechanism 1 mode the strain due to axial tension may be approximated by the following. Using the sketch of Figure 16 the increase in the halfspan length Δ may be approximated

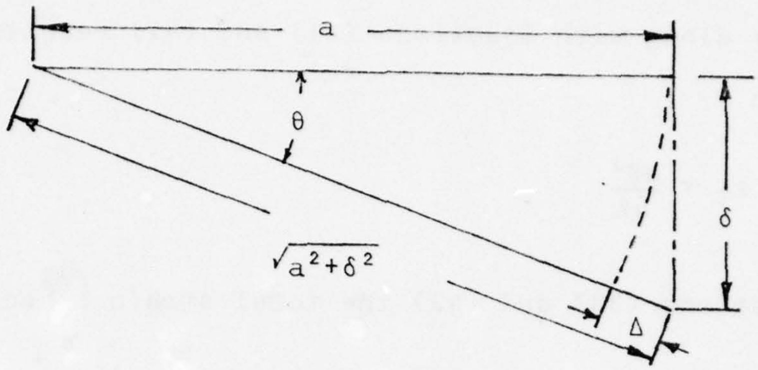


Figure 16. Sketch for approximating ε_t.

$$\Delta = (\sqrt{a^2+\delta^2})-a$$

$$= a\left(\sqrt{1+\frac{\delta^2}{a^2}}-1\right) \quad (40)$$

Expanding the radical in a binomial series Equation (40) becomes

$$\Delta = a\left(1+\frac{1}{2}\frac{\delta^2}{a^2}+\ldots-1\right)$$

$$\simeq \frac{\delta^2}{2a} \quad . \quad (41)$$

Assuming half the increase Δ takes place at the hinge and is uniform over the deformed length ℓ then

$$\varepsilon_t = \Delta/\ell \quad . \quad (42)$$

Approximating the rotation as $\theta \approx \delta/a$, which yields $\delta \approx \theta a$, and using this along with Equations (41) and (42) results in the strain

$$\varepsilon_t = \frac{a\theta^2}{4\ell} \quad . \tag{43}$$

Using Equations (39) and (43) the total strain ε becomes

$$\varepsilon = \frac{\theta a}{\ell}\left(\frac{\theta}{4}+\frac{d}{2a}\right) \quad . \tag{44}$$

Examining Equation (44) in terms of the ultimate strain of the reinforcing elements and solving for θ_u required to produce ultimate strain gives

$$\theta_u = \frac{d}{a}\left(\sqrt{\frac{4\ell a\varepsilon_u}{d^2}+1} -1\right) \quad . \tag{45}$$

For a mechanism 2 response the same argument may be used and the resulting equation for this response would be identical except the half span a would be replaced with x_h and the Equation (45) for a general case becomes

$$\theta_u = \frac{d}{x_h}\left(\sqrt{\frac{4\ell x_h\varepsilon_u}{d^2}+1} -1\right) \quad . \tag{46}$$

III. DISCUSSION AND RESULTS

3.1 Introduction

The previous sections contain the necessary equations of motion to determine the dynamic response of beams and plates for spatially and time varying loading functions for the assumed plastic hinge motion. The equations as derived, with adjustments of hinge moment term, would apply to thin metal structures as well as reinforced concrete structures but would not be applicable to very brittle material such as unreinforced concrete. The main underlying assumption is that the structure will take on the shape as proposed and portions of the structure respond almost as rigid body motion with localized failures only at hinges. Experiments [1,9] show that as a small explosive device is moved closer and closer to a much larger plate or beam localized failure of the concrete, i.e., actual separation of concrete from reinforcing elements, occurs and the structure simply does not contain the pressure. This type failure is a mechanism 4 type as described in Section 2.1. The onset of this type of failure appears to be dependent on the compressive strength of the concrete or the dynamic compressive pressure that the concrete can support before extreme cracking occurs. As a general rule an approximate dynamic load

factor of 1.5 to 2.0 is usually applied for many preliminary design problems and this type of thinking appears reasonable in this study. Based on this the upper bound of the maximum pressure of the spatial variations (P_C+P_E) should be set at approximately 1.5 times the compressive strength of the concrete. This rule is by no means meant to be a hard and fast rule but appears to be reasonable based on experimental observations[1,9] of both damage and pressure time histories.

The limit of rotation or center point deflection is based on the failure criterion discussed in Section 2.3.4. Large rotations displacements and strains occur and were accommodated in the basic assumptions. In the strain to fracture calculations the expression $\theta^2/2$ was considered to be small in comparison to unity. This more than likely should limit the rotation to something less than a third of a radian.

3.2 Numerical Analysis Procedure

The general equations as developed in Section 2.3 are rather complex and closed form solutions could be obtained given time. In fact closed form solutions are given for uniform loading of metal beams[4] and parabolic loadings of concrete beams[7]. However, for the nonlinear blast loading of plates the closed form solutions become very cumbersome and due to discontinuities in the solutions due to

assumed time functions and geometric response shape a numerical technique was developed to solve both plate and beam problems. The steps required for the solution are as follows:

1. Specify beam or plate problem.
2. Compilation of basic data on idealized structure: geometric size, pressure loadings, mechanical properties, position of load. (These become input parameters.)
3. Determine final hinge positions. For beams this is automatically the half span of the beam, and for plates this means solution of Equation (26) or (27) for z. The value of z determined then becomes a constant for the problem.
4. Determine the initial position of the hinge location as an indicator of which equation of motions are to be used. The initial condition of Equation (14) is used for this portion of the solution. For the beams Equation (14) is applied to Equations (10) and (12) or Equations (16). For the plates the initial condition is applied to Equations (29) and (31) or Equations (33) and (34).
5. From No. 4 above if $x_h = a, (X=1)$ then response is determined from a $\ddot{\theta}$ equation only. For beams the response is based on Equation (10) or (16a).

For plates the response is obtained using Equations (29) or (31). If $0<x_h<a$, $(0<X<1)$ then the initial response is governed by both the $\ddot{\theta}$ and $\ddot{\delta}$ equations. The response for beams is then determined from Equations (10) and (12) or Equations (16). For plates Equation (29) and (31) or Equations (33) and (34) will be used for $0<x_h<a$. If the initial response is $0<x_h<a$ then the hinge location moves toward the center position and when $x_h = a$ the solution reverts to the mechanism 1 mode of using only the $\ddot{\theta}$ equations.

6. In No. 5 above the proper response equations of motion are selected along with the correspoinding $\dot{WF}$, $\dot{WP}$, $\dot{PE}$ equations and they are solved simultaneously.
7. Calculate $\theta,\delta,\dot{\delta},WF,WP,PE,x_h,\theta_u$. Print data for current time, compare θ to θ_u and print asterisk if $\theta>\theta_u$.
8. Return to next case.

The numerical technique used in solving for the initial hinge location x_h for beams and plates and for the final hinge location zb for plates is a bisection (binary search) method[10] described in some detail in Section A.5.1 of the appendix. Since the rate of plastic work, rate of pressure work and rate of potential energy change are all functions of $\dot{\theta}$ and must be integrated with respect to time numerically they are

combined with the equations of motion and solved simultaneously using a 4th order Runge-Kutta technique[11]. This method is also discussed in detail in Section A.5.2 of the appendix.

3.3 Numerical Examples

3.3.1 Introduction

The numerical examples included here are response calculations of two test items of Reference 9. The test items as shown typically in Figure 17 are completely buried with soil before testing. The explosive is placed so as to be on a line normal to and at the center of the test face. The test items are essentially six sided boxes of 36x36x16in (.91x.91x.41m) and 48x36x16in (1.22x.91x.41m) with the side opposite the test face open. The side walls and the test face are 4 in (.1m) thick. For numerical analysis the test face was modelled as a 34x34x4in (.91x.91x.1m) plate for test case 1 and 44x34x4in (1.12x.91x.1m) for test case 2 with 1% tensile reinforcement ratio for both. Reinforcing elements are spaced 2 in (5.1cm) apart through the thickness and 1.0in (2.5cm) from the plate surface giving a reinforced distance d (see Figure 3) of 3.0in (7.6cm) for each slab.

Based on pressure measurements for tests of References 1 and 9 a linear pressure (β=0) of

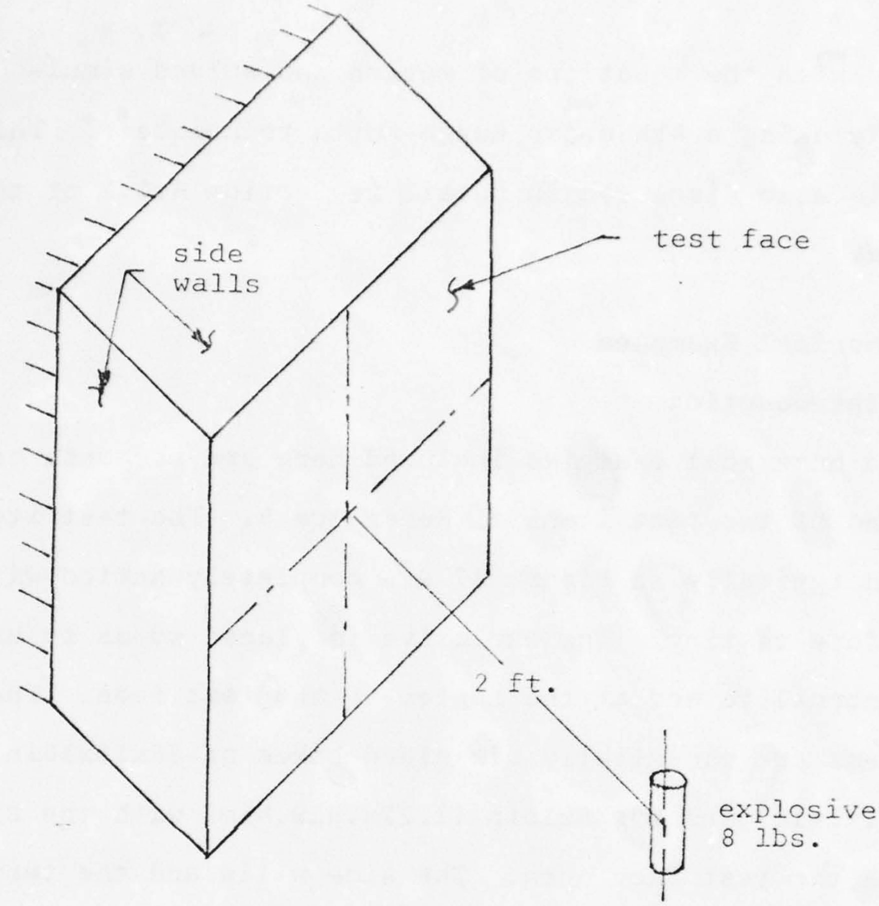


Figure 17. Schematic of test specimen used in numerical example.

$$P(x) = 2100 + 4300\frac{x}{a}\ \text{psi}$$
$$P(y) = 2100 + 4300\frac{y}{a}\ \text{psi} \tag{47}$$

was used for the spatial loading of each slab. A general time function, with $\alpha = 0$, of

$$f(t) = (1 - t/.5\times10^{-3}) \tag{48}$$

was used for the pressure time variation. The general shapes of the spatial variation and time variation are shown in Figure 18. From Reference 9 the concrete compressive strength was

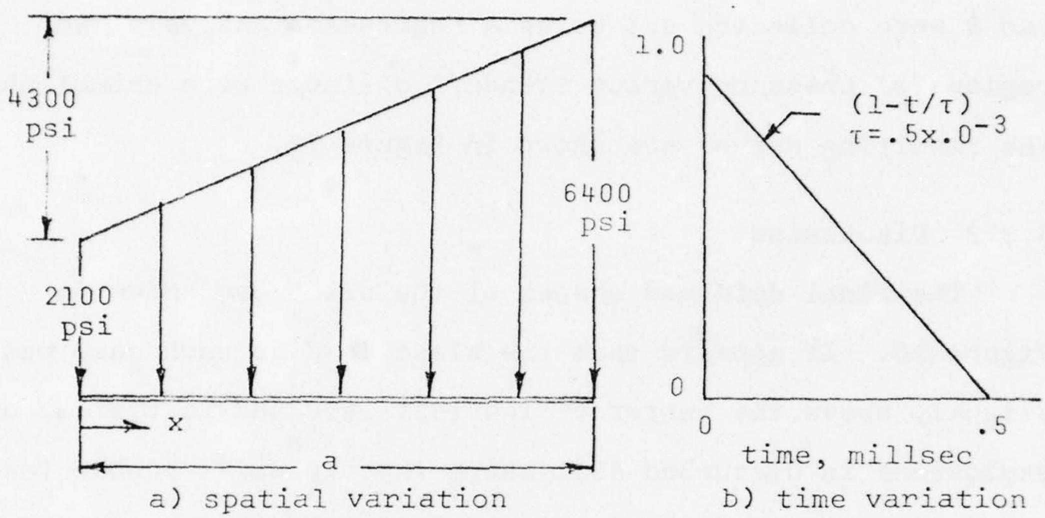


Figure 18. Pressure loading for numerical example.

6000 psi (41.4MPa) and the reinforcement strength was 70,000 psi (483.6 MPa). The time step (TINCR) for the numerical analysis was chosen as 0.5×10^{-4} sec with a data print frequency (TPRINT) every 0.2×10^{-3} sec and a maximum real time response (TMAX) of 0.1 sec.

3.3.2 Results

The results of the two numerical examples are shown in Tables I and II. For the tabulated results of TIME, THETA, etc., the time is given only in the upper left hand column.

The rows in the lower columns PRESSURE WORK, PLASTIC WORK, etc., correspond to the rows of the TIME column. The pressure loadings used for the numerical examples were taken from References 1 and 9. All the pressure data for References 1 and 9 were collected and using a regression analysis two regions of pressure versus standoff distance were established. The resulting curves are shown in Figure 19.

3.3.3 Discussion

The final deformed shapes of the slabs are shown in Figure 20. It appears that the blast load in each case was slightly above the center of the test face and is typical of explosions in disturbed soil where the explosive bubble tends to rise into the top loose soil. The analysis showed the initial hinge location was approximately 11.5 in (29.2cm) from the edge for each slab. In a cross section across the center line of the slab, the deformed shape after some displacement has occurred would be similar to the sketch of Figure 21. The rotation occurs at A and C of Figure 21 and tensile cracks form in the concrete at and near A and compression cracks and dislodged material occurs at the hinge at C. In general this appears to have occurred in both test specimens of Figure 20. The analysis predicts a maximum deflection of 4.4in (11.2cm) with a maximum experimental

TABLE I. RESULTS OF NUMERICAL EXAMPLE NO. 1

```
                     CALCULATIONS ON A CONCRETE PLATE
    TEST CASE 1      34 X 34 X 4     LINEAR

INPUT VALUES

    PLATE HALF WIDTH OR BEAM HALF SPAN, IN.                  (A)  17.000000
    BEAM OR PLATE THICKNESS, IN.                             (H)  4.0000000
    LENGTH TO WIDTH RATIO, DIMENSIONLESS                    (AF)  1.0000000
    MASS PER UNIT AREA, LBS-SEC.SQD/IN.CUBED                (XM)  .89960000E-03

    MAXIMUM DISTRIBUTED PRESSURE LOAD, PSI.                 (PC)  4300.0000
    UNIFORM PRESSURE LOAD, PSI.                             (PE)  2100.0000
    PRESSURE DECAY, DIMENSIONLESS                        (ALPHA) 0.
    PRESSURE DURATION, SEC.                                (TAU)  .50000000E-03

    SUPPORT FACTOR: 1=SIMPLY, 2=CLAMPED                      (F)  2.0000000
    WAVE FUNCTION: 1=GENERAL, 2=SQUARE                  (WAVEFN)  1.0000000
    WEIGHT VECTOR: 0=VERT, 1=EXP BLW, -1=EXP ABV (AIDA) 0.
    SPATIAL PRESSURE DECAY CONSTANT,DIMENSIONLESS(BETA) 0.

    CONCRETE COMPRESSIVE STRENGTH, PSI.                 (SIGMAC)  6000.0000
    REINFORCED STEEL YIELD STRESS, PSI.                 (SIGMAR)  70000.000
    REINFORCEMENT RATIO IN TENSION, DIMENSIONLESS            (O)  .10000000E-01
    REINFORCING DISTANCE, IN.                                (D)  3.0000000

    TIME INCREMENT, SEC.                                 (TINCR)  .50000000E-04
    TIME MAXIMUM, SEC.                                    (TMAX)  .10000000
    TIME STEP INTERVAL PER PRINTED LINE, SEC.  (TPRINT)  .20000000E-03

COMPUTED CONSTANT VALUES

    PLATE HALF LENGTH, IN.                                   (B)  17.000000
    RATIO OF FINAL HINGE LOC TO B, DIMENSIONLESS             (Z)  1.0000000
    HINGE MOMENT, IN.-LBS./IN.                             (XMU)  5279.7150
    WEIGHT PER UNIT AREA, LBS./IN.SQ.                        (W)  .34760544
    ORIGINAL HINGE LOCATION, IN.                            (XH)  11.542934

        CLAMPED-SUPPORTED              GENERAL TIME FUNCTION


         VERTICAL WALL                           LINEAR LOAD
```

TABLE I. RESULTS OF NUMERICAL EXAMPLE NO. 1 (Concluded)

CALCULATIONS ON A CONCRETE PLATE
TEST CASE 1 34 X 34 X 4 LINEAR

CLAMPED-SUPPORTED GENERAL TIME FUNCTION

TIME (SECONDS)	THETA (RADIANS)	MIDPT. VEL. (IN./SEC.)	MIDPT. DELTA (INCHES)
0.	0.	0.	0.
.20000000E-03	.90191178E-02	976.80166	.10572560
.40000000E-03	.29957758E-01	1468.2165	.35836334
.60000000E-03	.53901490E-01	1530.0061	.66230337
.80000000E-03	.76376661E-01	1530.0061	.96830459
.10000000E-02	.97282774E-01	1530.0061	1.2743058
.12000000E-02	.11670349	1530.0061	1.5803070
.14000000E-02	.13471441	1530.0061	1.8863082
.16000000E-02	.15138235	1530.0061	2.1923095
.18000000E-02	.16676520	1530.0061	2.4983107
.20000000E-02	.18091173	1530.0061	2.8043119
.22000000E-02	.19386114	1530.0061	3.1103131
.24000000E-02	.20564225	1530.0061	3.4163143
.26000000E-02	.21627202	854.78575	3.6766243
.28000000E-02	.22575492	757.30824	3.8378337
.30000000E-02	.23409103	659.83073	3.9795476
.32000000E-02	.24128035	562.35322	4.1017659
.34000000E-02	.24732287	464.87571	4.2044888
.36000000E-02	.25221850	367.39820	4.2877162
.38000000E-02	.25596754	269.92069	4.3514481
.40000000E-02	.25856958	172.44317	4.3956845
.42000000E-02	.26002502	74.965665	4.4204254

PRESSURE WORK (IN.-LBS.)	PLASTIC WORK (IN.-LBS.)	KINETIC ENERGY (IN.-LBS.)	HINGE LOCATION (INCHES)
0.	0.	0.	11.542934
177758.94	12952.197	164806.74	11.722388
395472.30	43021.825	352450.47	11.962284
427914.43	77406.985	350507.44	12.287292
427914.43	109683.18	318231.24	12.678017
427914.43	139706.09	288208.34	13.098987
427914.43	167595.84	260318.59	13.541215
427914.43	193461.00	234453.43	14.002276
427914.43	217397.54	210516.89	14.481936
427914.43	239488.59	188425.84	14.981007
427914.43	259804.17	168110.26	15.500995
427914.43	278400.59	149513.84	16.044025
427914.43	295319.24	132595.19	16.612901
427914.43	310584.45	117329.98	17.000000
427914.43	324202.69	103711.74	17.000000
427914.43	336174.03	91740.397	17.000000
427914.43	346498.48	81415.945	17.000000
427914.43	355176.04	72738.383	17.000000
427914.43	362206.72	65707.713	17.000000
427914.43	367590.49	60323.933	17.000000
427914.43	371327.38	56587.045	17.000000
427914.43	373417.38	54497.048	17.000000

MAXIMUM DEFLECTION = 4.4256708 AT TIME = .44000000E-02

AN ASTERISK INDICATES THAT A REINFORCING ELEMENT HAS FRACTURED

TABLE II. RESULTS OF NUMERICAL EXAMPLE NO. 2

```
                     CALCULATIONS ON A CONCRETE PLATE
    TEST CASE 2      44 X 34 X 4       LINEAR

INPUT VALUES

    PLATE HALF WIDTH OR BEAM HALF SPAN, IN.                  (A)  17.000000
    BEAM OR PLATE THICKNESS, IN.                             (H)  4.0000000
    LENGTH TO WIDTH RATIO, DIMENSIONLESS                    (AR)  1.2900000
    MASS PER UNIT AREA, LBS-SEC.SQD/IN.CUBED                (XM)  .89960000E-03

    MAXIMUM DISTRIBUTED PRESSURE LOAD, PSI.                 (PC)  4300.0000
    UNIFORM PRESSURE LOAD, PSI.                             (PE)  2100.0000
    PRESSURE DECAY, DIMENSIONLESS                        (ALPHA)  0.
    PRESSURE DURATION, SEC.                                (TAU)  .50000000E-03

    SUPPORT FACTOR: 1=SIMPLY, 2=CLAMPED                      (F)  2.0000000
    WAVE FUNCTION: 1=GENERAL, 2=SQUARE                  (WAVEFN)  1.0000000
    WEIGHT VECTOR: 0=VERT, 1=EXP BLW, -1=EXP ABV (AIDA) 0.
    SPATIAL PRESSURE DECAY CONSTANT,DIMENSIONLESS(BETA) 0.

    CONCRETE COMPRESSIVE STRENGTH, PSI.                (SIGMAC)  6000.0000
    REINFORCED STEEL YIELD STRESS, PSI.                (SIGMAR)  70000.000
    REINFORCEMENT RATIO IN TENSION, DIMENSIONLESS           (Q)  .10000000E-01
    REINFORCING DISTANCE, IN.                               (D)  3.0000000

    TIME INCREMENT, SEC.                                (TINCR)  .50000000E-04
    TIME MAXIMUM, SEC.                                   (TMAX)  .10000000
    TIME STEP INTERVAL PER PRINTED LINE, SEC.          (TPRINT)  .20000000E-03

COMPUTED CONSTANT VALUES

    PLATE HALF LENGTH, IN.                                   (B)  21.930000
    RATIO OF FINAL HINGE LOC TO B, DIMENSIONLESS             (Z)  .87852855
    HINGE MOMENT, IN.-LBS./IN.                             (XMU)  5279.7150
    WEIGHT PER UNIT AREA, LBS./IN.SQ.                        (W)  .34760544
    ORIGINAL HINGE LOCATION, IN.                            (XH)  11.466199

        CLAMPED-SUPPORTED               GENERAL TIME FUNCTION

        VERTICAL WALL                            LINEAR LOAD
```

TABLE II. RESULTS OF NUMERICAL EXAMPLE NO. 2 (Concluded)

CALCULATIONS ON A CONCRETE PLATE

TEST CASE 2   44 X 34 X 4   LINEAR

CLAMPED-SUPPORTED   GENERAL TIME FUNCTION

TIME (SECONDS)	THETA (RADIANS)		MIDPT. VEL. (IN./SEC.)	MIDPT. DELTA (INCHES)
0.	0.		0.	0.
.20000000E-03	.88594534E-02		951.05885	.10295644
.40000000E-03	.29497509E-01		1428.9726	.34888196
.60000000E-03	.53255951E-01		1488.9974	.64467941
.80000000E-03	.75781298E-01		1488.9974	.94247888
.10000000E-02	.96962593E-01		1488.9974	1.2402784
.12000000E-02	.11686975		1488.9974	1.5380778
.14000000E-02	.13556763		1488.9974	1.8358773
.16000000E-02	.15311561		1488.9974	2.1336768
.18000000E-02	.16956757		1488.9974	2.4314762
.20000000E-02	.18497211		1488.9974	2.7292757
.22000000E-02	.19937269		1488.9974	3.0270752
.24000000E-02	.21280777		1488.9974	3.3248747
.26000000E-02	.22531088		1488.9974	3.6226741
.28000000E-02	.23691061		1488.9974	3.9204736
.30000000E-02	.24763242		874.65764	4.2097512
.32000000E-02	.25749121		801.33661	4.3773506
.34000000E-02	.26648740		728.01558	4.5302858
.36000000E-02	.27462099		654.69455	4.6685568
.38000000E-02	.28189198		581.37352	4.7921636
.40000000E-02	.28830037		508.05249	4.9011062
.42000000E-02	.29384615	*	434.73146	4.9953846
.44000000E-02	.29852934	*	361.41043	5.0749988
.46000000E-02	.30234993	*	288.08940	5.1399488
.48000000E-02	.30530792	*	214.76837	5.1902346
.50000000E-02	.30740330	*	141.44734	5.2258561
.52000000E-02	.30863609	*	68.126307	5.2468135

PRESSURE WORK (IN.-LBS.)	PLASTIC WORK (IN.-LBS.)	KINETIC ENERGY (IN.-LBS.)	HINGE LOCATION (INCHES)
0.	0.	0.	11.466199
230559.15	13819.477	216739.67	11.621083
513725.16	46011.884	467713.28	11.827506
556082.63	83071.648	473010.99	12.105303
556082.63	118207.95	437874.67	12.436827
556082.63	151247.74	404834.89	12.791308
556082.63	182300.05	373782.58	13.160616
556082.63	211466.06	344616.57	13.542151
556082.63	238838.39	317244.24	13.935071
556082.63	264501.09	291581.54	14.339276
556082.63	288529.97	267552.66	14.755066
556082.63	310992.80	245089.83	15.182998
556082.63	331949.60	224133.03	15.623841
556082.63	351452.67	204629.96	16.078558
556082.63	369546.58	186536.05	16.548324
556082.63	386271.07	169811.56	17.000000
556082.63	401649.37	154433.26	17.000000
556082.63	415682.13	140400.50	17.000000
556082.63	428369.37	127713.27	17.000000
556082.63	439711.07	116371.57	17.000000
556082.63	449707.23	106375.40	17.000000
556082.63	458357.87	97724.766	17.000000
556082.63	465662.97	90419.665	17.000000
556082.63	471622.54	84460.097	17.000000
556082.63	476236.57	79846.061	17.000000
556082.63	479505.07	76577.559	17.000000
556082.63	481428.04	74654.589	17.000000

MAXIMUM DEFLECTION = 5.2531067   AT TIME = .54000000E-02

AN ASTERISK INDICATES THAT A REINFORCING ELEMENT HAS FRACTURED

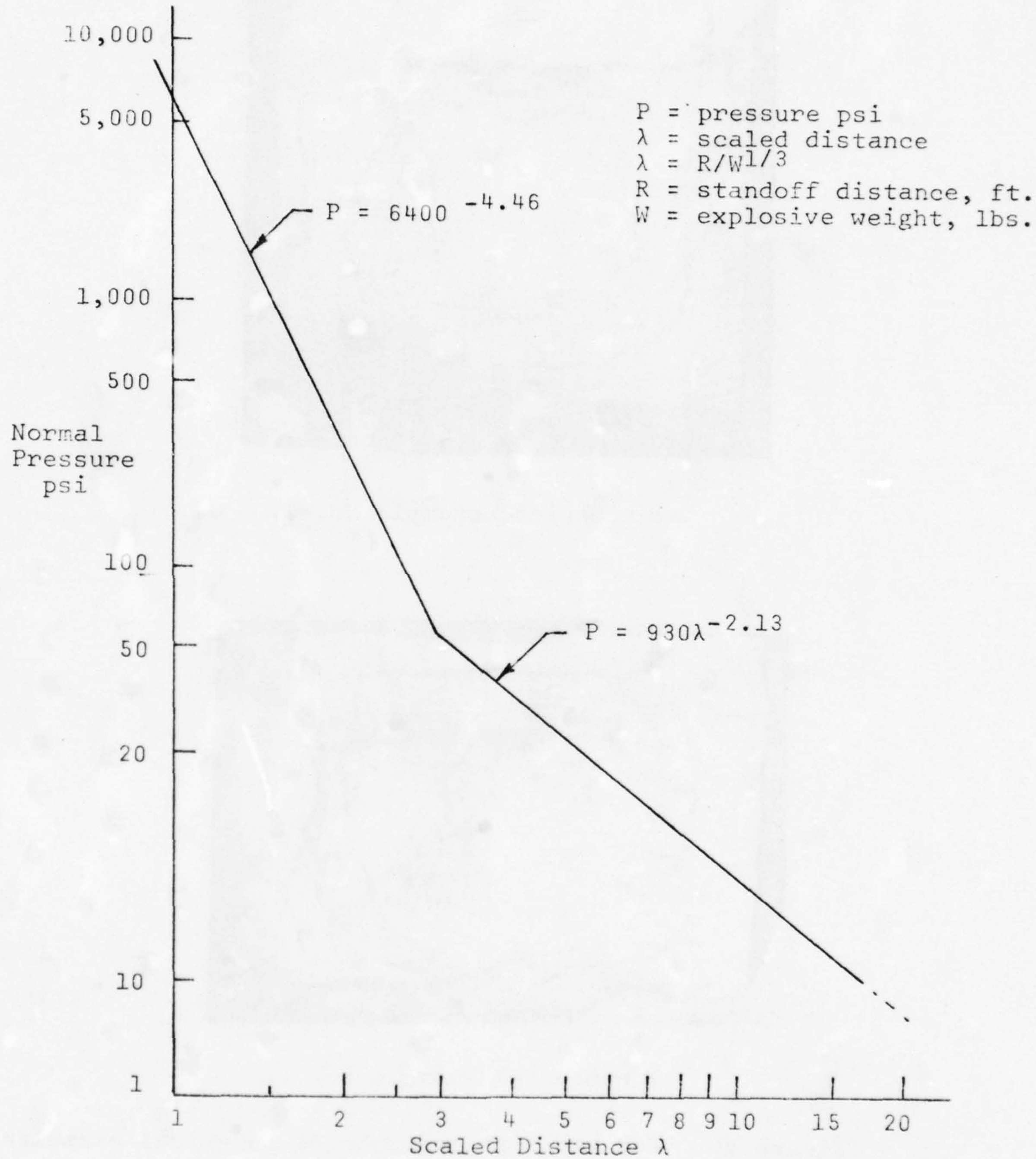
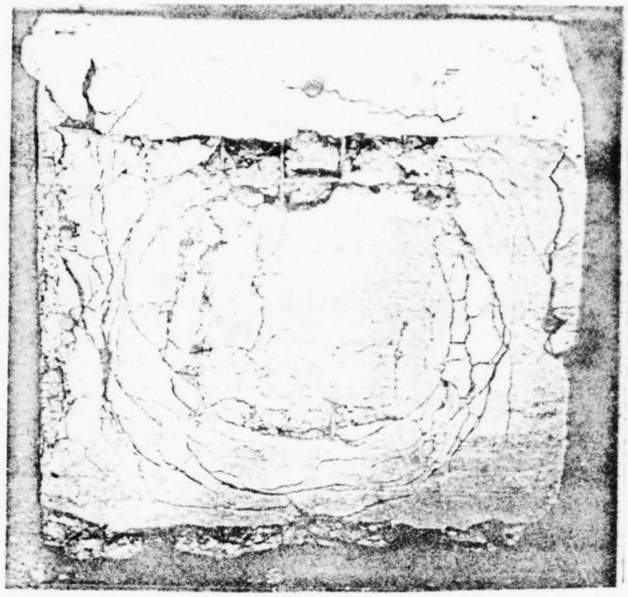
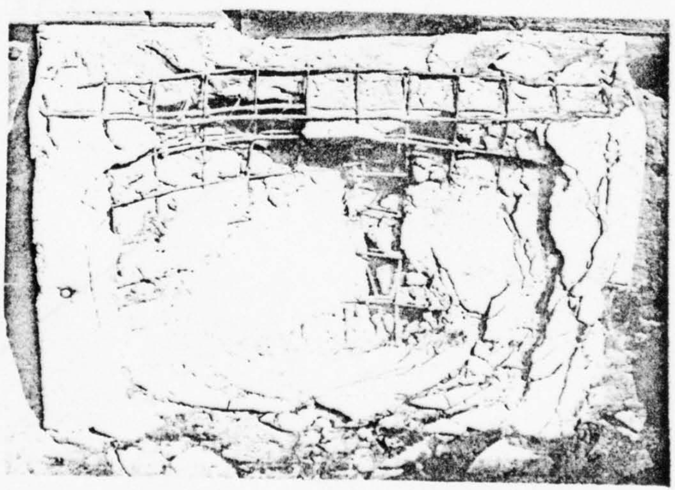


Figure 19. Pressure versus scaled distance.



a) numerical example No. 1.



b) numerical example No. 2.

Figure 20. Post-test photographs of numerical examples.

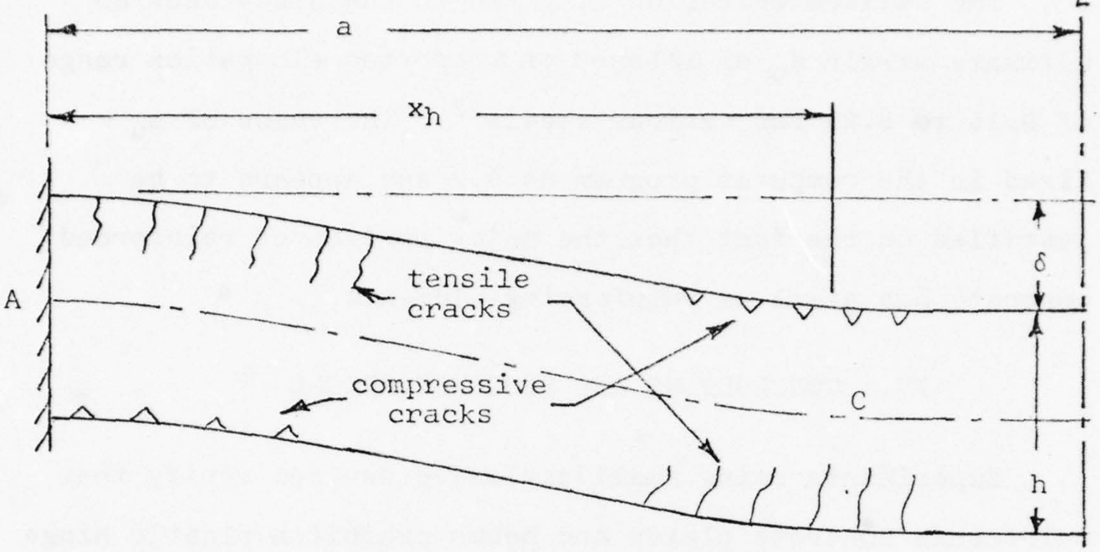


Figure 21. Schematic of plate cross section through middle of plate.

deflection of 2.8in (7.11cm) for test case 1. For test case 2 the analysis shows a maximum displacement of 5.25in (13.3cm) and the experimentally observed maximum deflection was approximately 8in (20.3cm) at the center. The analysis indicates failure of a reinforcing element occurred in test case 2 and not of 1 and Figure 20 shows evidence of severe damage of test case 2 and not of test case 1. Probably the most important result of these analyses is that the analysis distinguishes between the severity of damage with just changes in the aspect ratio $\overline{AR}$ of the slab.

The failure criterion as given in the study uses an ultimate strain ε_u of 0.2 based on a reported elongation range of 0.15 to 0.25 for various steels[12]. The value of ε_u was fixed in the computer program as 0.2 and appears to be justified on the fact that the major portion of reinforced concrete has steel as reinforcing elements.

IV. CONCLUSIONS AND RECOMMENDATIONS

Experiments using small explosive devices verify that reinforced concrete plates and beams exhibit a plastic hinge response when subjected to dynamic loadings of such devices.

Using failure mechanisms based on plastic hinge response, equations of motion, for reinforced concrete beams and plates, are derivable and their solutions predict responses that are in reasonable agreement with experiments. These solutions are limited to simply supported and fixed edge conditions and their accuracy is heavily dependent on the accuracy of the pressure time loading function. The basic assumptions of these analyses, i.e., plastic hinge response, neglecting elastic motion and tensile concrete strength, coupled with the variations found in pressure loading predictions could produce overall response errors of ±50%.

In addition application of the derived equations of motion are limited to pressure loadings between the pressure

required for static collapse and pressures where the concrete begins to break up locally before the plastic hinge response begins.

It is recommended that the study of blast response of reinforced concrete structures be continued to include non-symmetric loadings, response to very localized or very near field explosives and complete shearing of small structural elements at boundaries (Mechanisms 3 and 4 of Section 2.1).

V. REFERENCES

1. Fuehrer, H. R. and Kesser, J. W., "Response of Buried Concrete Slabs to Underground Explosions," AFATL-TR-77-115, USAF Armament Lab., Eglin AFB, FL., September 1977.
2. Szilard, R., Theory and Analysis of Plates, Prentice Hall, Inc., 1974, pp. 511-612.
3. Timoshenko, S. P., and Gere, J. M., Mechanics of Materials, Van Nostrand Reinholt Co., 1972, pp. 289-316.
4. Abrahamson, G. R., Florence, A. L., and Lindberg, H. E., "Radiation Damage Study (RADS) - Vol. XIII, Dynamic Response of Beams, Plates and Shells," BSDTR-66-372, Vol. XIII, Ballistics System Div., Norton AFB, CA., September 1966.

5. Jones, N., "Plastic Failure of Ductile Beams Loaded Dynamically," J. of Eng. for Industry, Trans. ASME Vol. 98, Series B, No. 1, February 1976, pp. 131-136.

6. Menkes, S. B., and Opat, H. J., "Broken Beams," Experimental Mechanics, Vol. 13, 1973, pp. 480-486.

7. Florence, A. L., "Critical Loads for Reinforced Concrete Bunkers," DNA 4469F, Defense Nuclear Agcy, Washington, D. C., November 1977.

8. Ross, C. A., Nash, P. T., and Griner, G. R., "Failure of Underground Concrete Structures Subjected to Blast Loadings," Shock and Vibration Symposium, October 1978, Washington, D. C. To be published in 49th Shock and Vibration Bulletin.

9. Fuehrer, H. R. and Kesser, J. W., "Breach of Buried Concrete Structures," AFATL-TR-78-92, USAF Armament Lab., Eglin AFB, FL., August 1978.

10. LaFara, R. L., Computer Methods for Science and Engineering, Hayden Book Company, 1973, pp. 82-84.

11. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill Book Company, 1956, pp. 236-239.

12. ASM Metals Handbook, American Society of Metals, 1960.

APPENDIX

COMPUTER PROGRAM SPECIFICATION FOR CONCRE

A.1 Introduction

The following describes the computer program named CONCRE, Reinforced Concrete Beam or Plate Center Deflection. This program was developed and used to test the theory explained in the earlier part of this document. For a listing, see Section A.6.

The program, CONCRE, computes the deflection of the center point and edge rotation of a reinforced concrete beam or plate caused by a linear or blast load. The main functions of the program are DRIVER, NEXT-CASE, TIME-CONTROL, TIME-ZERO, BISECTION, TIME-STEP, RUNGE-KUTTA, CHECK-HINGE, and PRINT-ROUTINE. See Diagram A.2.1 for a description of the interaction of these functions.

This program was developed using the FORTRAN Extended Compiler and the NOS/BE1 Operating System on the CDC 6600 Computer at Eglin Air Force Base, Florida. It was written in a top-down, structured manner, containing numerous comments. It uses Standard FORTRAN, with the exception of the use of asterisks (*) to define Hollarith data within FORMAT statements and calls to certain system library functions. The program is complete in itself.

The input to the program should be in eighty-column card format. The only output is a printed listing, which includes the input as read, the results as requested and computed, and error messages, if any. For a further description of the input, see Section A.4.1. Section A.4.2 further describes the output. A sample run complete with input and output is given in Section 3.3.

A.2 Functional Description

The nine main functions of the program, CONCRE, are described in the subparagraphs below. The interaction between these functions is shown in Figure A.2.1. An abbreviated name is given for each function for clarity and simplification in the diagrams. In most cases, these abbreviated names match the name of one of the program's subprocedures, although a function may be made up of more than one procedure. Figure A.2.2 shows the relationship between the 21 subprocedures. For further information on each function, see the comments embedded in the listing (Section A.6) for each procedure within the function.

A.2.1 Driver

The function, DRIVER, drives the program taking care of initialization and clean up activities as well as controlling the main flow of the program. This is the main function of

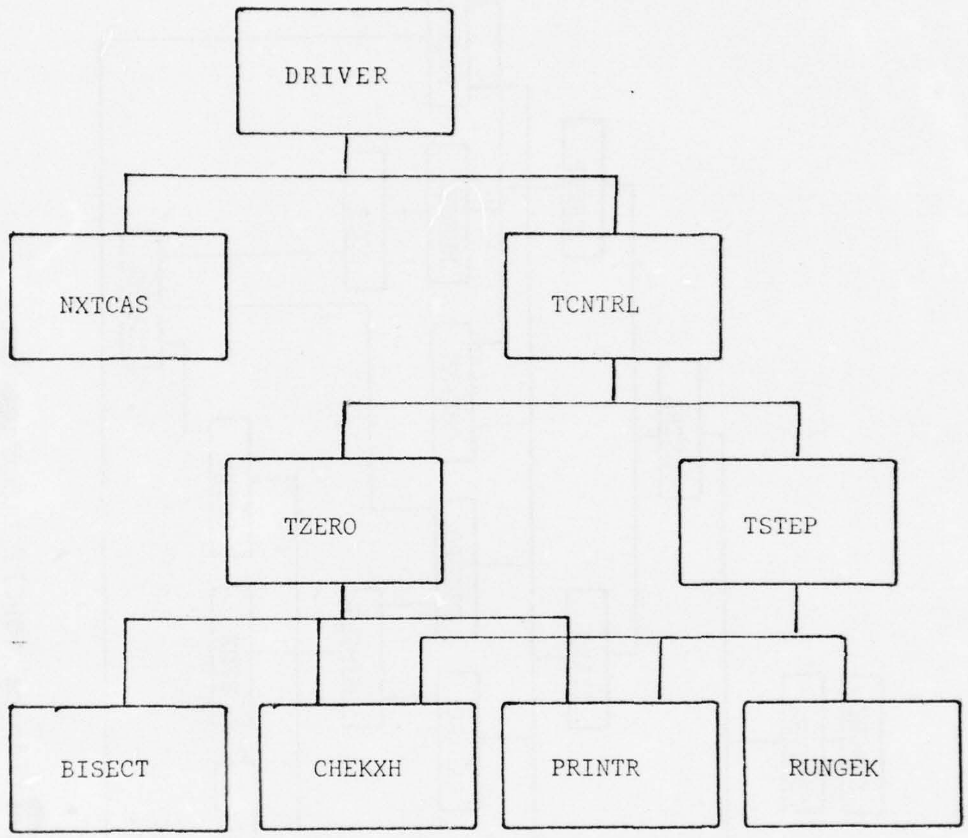


Figure A.2.1. Relationships between CONCRE functions

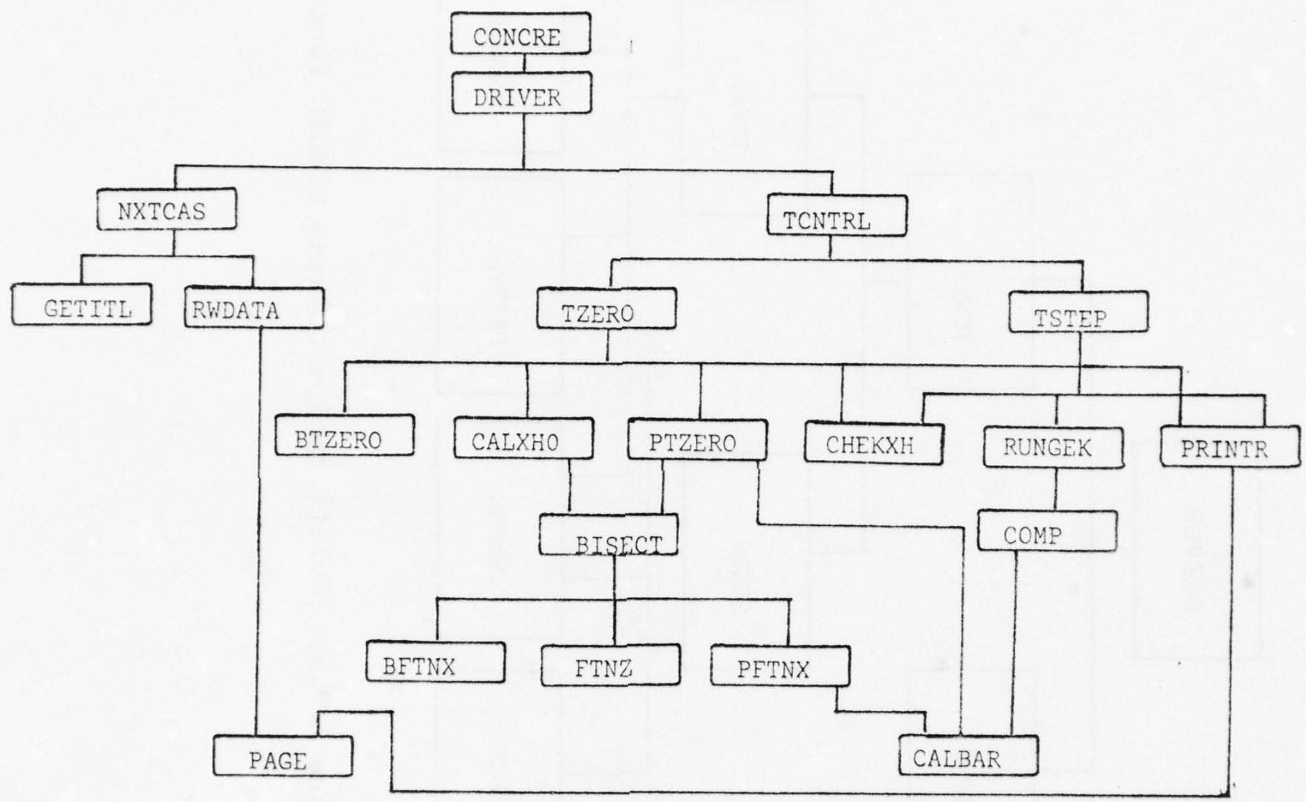


Figure A.2.2. Interaction of CONCRE Subprocedures

the program and begins when the program begins execution. It is terminated when an End-of-File condition is returned by function NEXT-CASE.

This function is made up of the two FORTRAN procedures, CONCRE and DRIVER. It calls the functions, NEXT-CASE and TIME-CONTROL. Flow Diagrams of the two procedures, CONCRE and DRIVER, are shown in Figures A.2.1.1 and A.2.1.2.

A.2.2 Next-Case

The function, NEXT-CASE (or NXTCAS), finds the next case in the stream of input, reads, stores, and prints out the data associated with that case. If errors are detected in the input data, proper error messages are printed, and the Input-Error-Flag is returned to the DRIVER function as set so that this case will be terminated and NEXT-CASE recalled. Should the End-of-File be found before finding the first card of the next case, the function returns control to the DRIVER function for a normal termination. Should the End-of-File be found before reading the last data card of the next case, error messages will be printed before return to the DRIVER function.

This function, NEXT-CASE, is called by and returns to the DRIVER function. It sets the Input-Error-Flag and End-of-File-Flag used by the DRIVER function. It calls no other

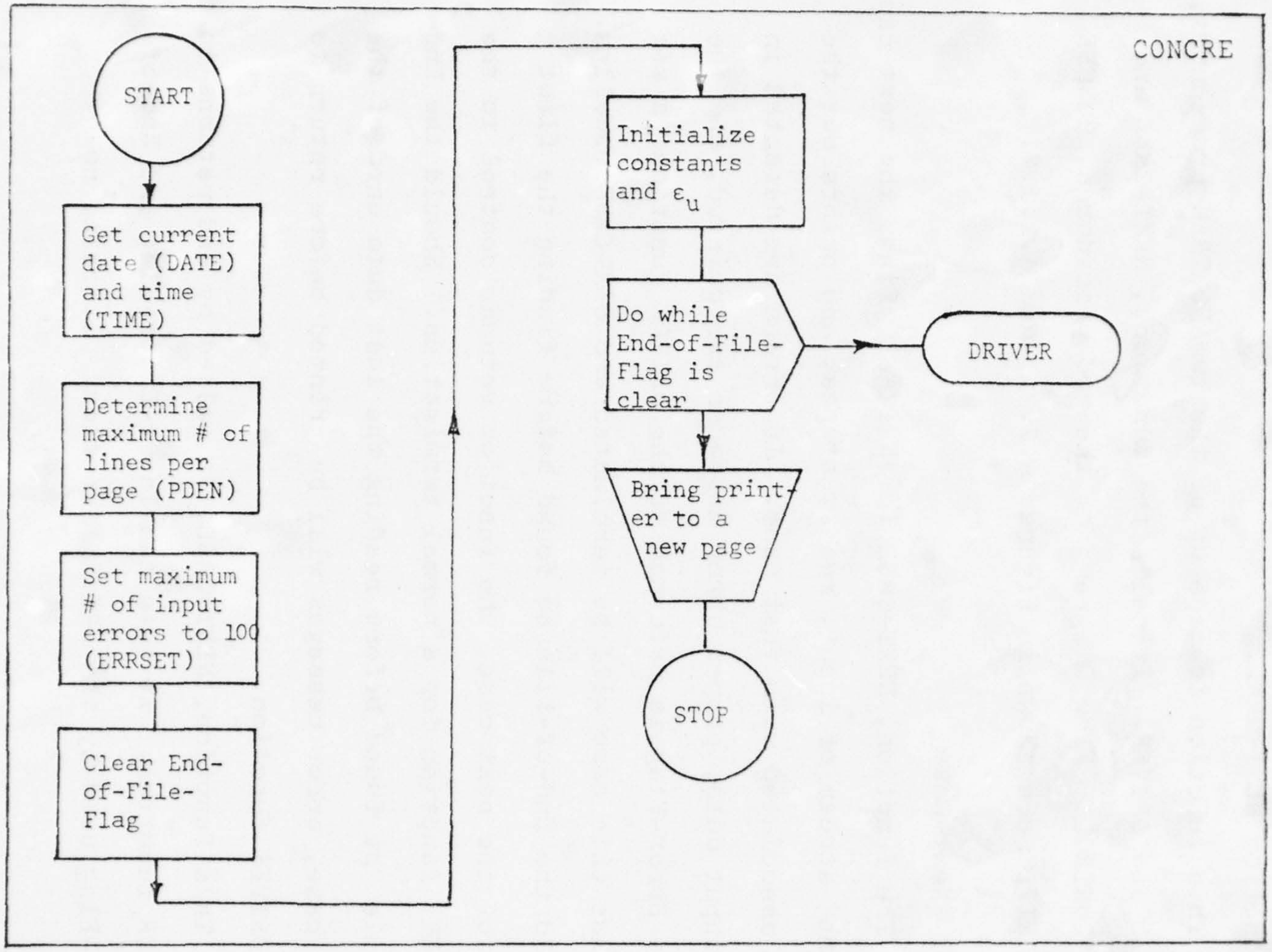


Figure A.2.1.1. Flow Diagram of the Main Procedure, CONCRE

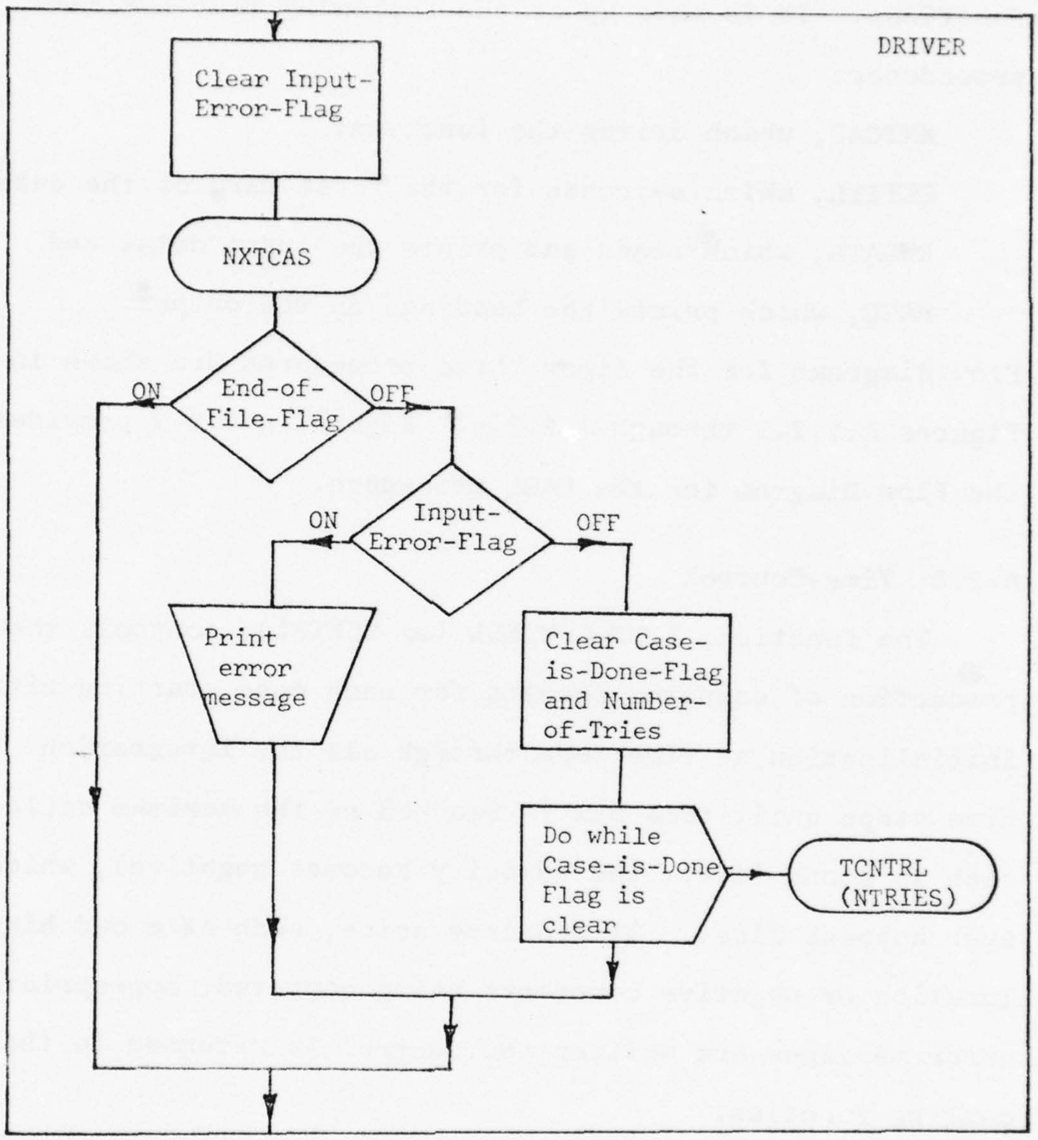


Figure A.2.1.2. Flow Diagram of Subroutine DRIVER

functions. It is made up of the following four FORTRAN procedures:

NXTCAS, which drives the function;

GETITL, which searches for the first card of the case;

RWDATA, which reads and prints the input data; and

PAGE, which prints the headings on the output.

Flow diagrams for the first three procedures are shown in Figures A.2.2.1 through A.2.2.3. Figure A.2.9.2 provides the Flow Diagram for the PAGE procedure.

A.2.3 Time-Control

The function, TIME-CONTROL (or TCNTRL), controls the production of computed results for each case starting with initialization at time zero through all the integration time steps until time-max is reached or the maximum deflection is found (i.e., the velocity becomes negative), whichever happens first. If problems arise, such as a bad hinge location or negative constants being computed, appropriate error messages are written and control is returned to the calling function.

This function, TIME-CONTROL, is called by and returns to the DRIVER function. It sets the Case-is-Done-Flag if computation is completed, if disasterous errors are found, or if the case has been tried a maximum number of times.

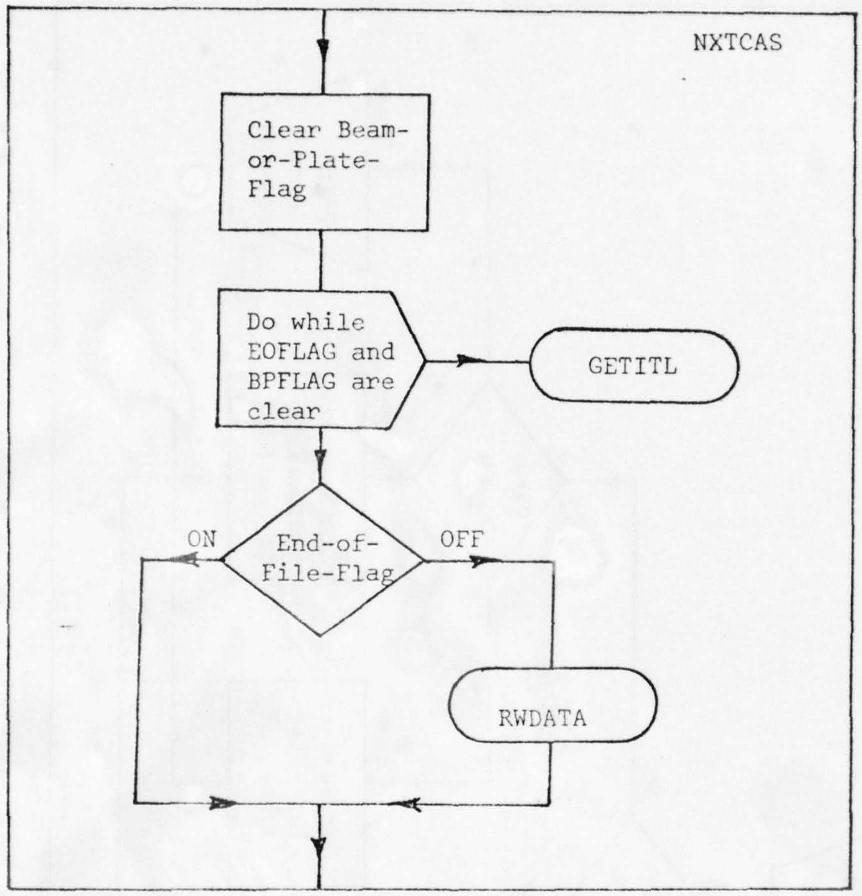


Figure A.2.2.1. Flow Diagram of Subroutine NXTCAS

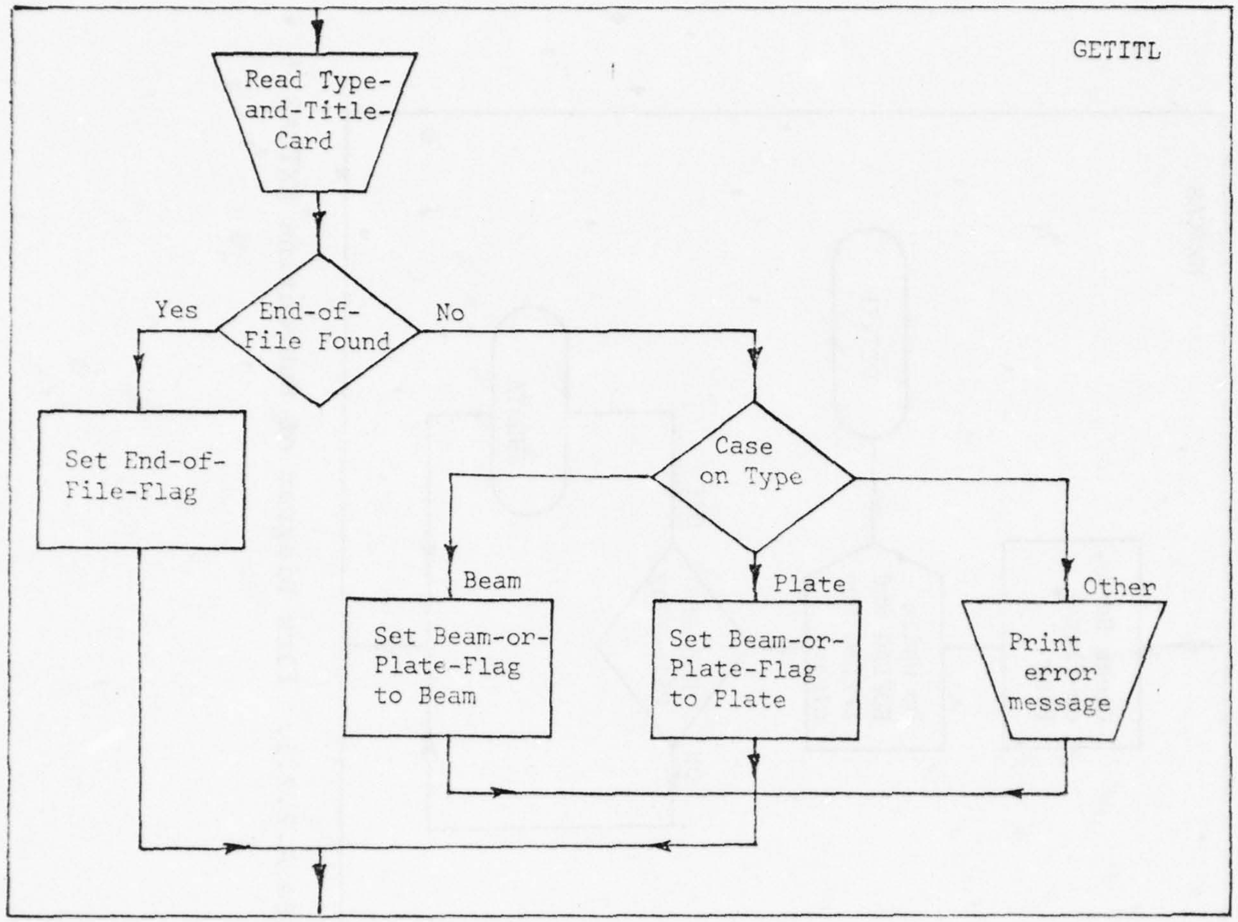


Figure A.2.2.2. Flow Diagram of Subroutine GETITL

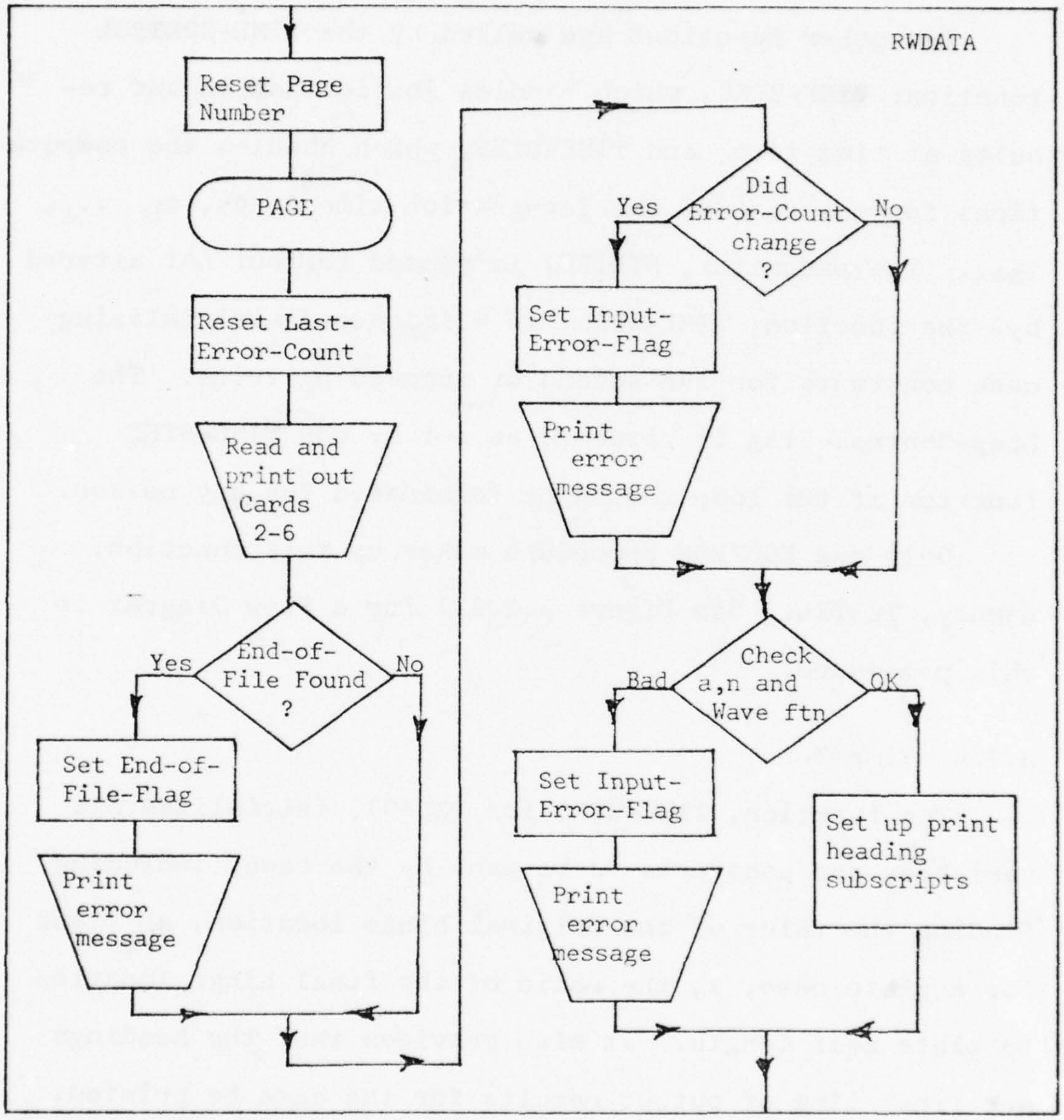


Figure A.2.2.3. Flow Diagram of Subroutine RWDATA

The parameter, NTRIES, which is returned to the calling function, counts the number of times the current case has called this function, TIME-CONTROL.

Two other functions are called by the TIME-CONTROL function; TIME-ZERO, which handles initialization and results at time zero; and TIME-STEP, which handles the computations for the rest of the integration time steps, t_1, ..., t_{max}. The parameter, NTRIES, is passed to (but not altered by) the function, TIME-ZERO, to eliminate re-initializing case constants for the second or succeeding tries. The Loop-Control-Flag is returned as set by the TIME-STEP function if the loop should be terminated for any reason.

Only one FORTRAN procedure makes up this function; namely, TCNTRL. See Figure A.2.3.1 for a Flow Diagram of this procedure.

A.2.4 Time-Zero

The function, TIME-ZERO (or TZERO), initializes all variables and constants to be used by the case, including finding the value of the original hinge location, x_{h_0}, and for a plate case, z, the ratio of the final hinge location to plate half length. It also provides that the headings and first line of output results for the case be printed. Should invalid constants (i.e., negative) be calculated, or,

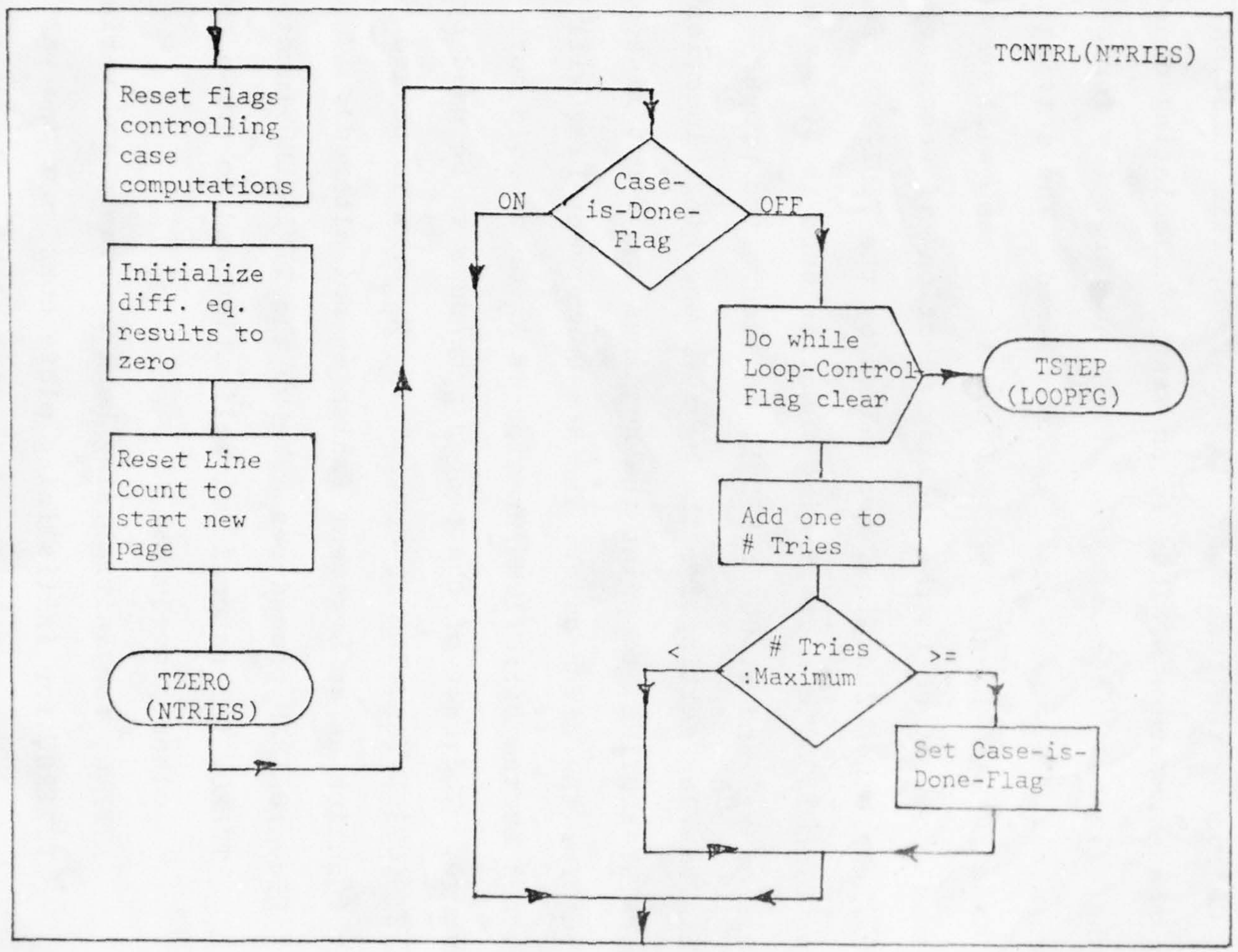


Figure A.2.3.1. Flow Diagram of Subroutine TCNTRL.

for a plate case, a value of z not converge, the Case-is-Done-Flag will be set.

Called by the function, TIME-CONTROL, this function, uses the parameter, NTRIES, to prevent recalculating constants or the original hinge location after the first try for a given case. This parameter is not altered. The setting of the Case-is-Done-Flag as above assures the case will be terminated upon return to the calling function, if necessary.

Three other functions are called by the TIME-ZERO function: BISECTION, a root-finding function used to find the values of x_{h_0} and z; CHECK-HINGE, to assure the proper Mechanism-Flag setting for the newly-found hinge location, and PRINT-ROUTINE, to print the headings and first line of output for the given case. The Non-Convergent-Flag will be returned by the BISECTION function if X_0 or z would not converge. The name of the proper procedure to be used by the BISECTION function in determining X_0 or z is passed to that function as an argument (which is not altered).

Five FORTRAN procedures make up the TIME-ZERO function:

TZERO, for general initialization and to drive the other procedures;

BTZERO, for initializing beam constants and variables;

PTZERO, for initializing plate constants and variables;

CALXHO, for deriving the original hinge location; and

CALBAR, to assist in initializing plate constants for mechanism 1 Runge-Kutta computations.

Flow Diagrams of these first four procedures will be found in Figures A.2.4.1 through A.2.4.4. The Flow Diagram for the procedure, CALBAR, will be found in Figure A.2.7.3.

A.2.5 Bisection

The function, BISECTION (or BISECT) is a root-finding technique. Using the BISECTION (or Binary Search) Method, this function derives the root of one of three different formulas to find the values of X_0 and z. If the root cannot be found or does not converge, the Error-Flag (Non-Convergent-Flag) is returned to the calling function.

Called by the TIME-ZERO function, this function uses parameters to define the likely interval within which the root should be and the formula to be used. It returns the root, if found, and the Error-Flag, if set, to the calling function. The BISECTION function calls no other function.

This function consists of five FORTRAN procedures:

BISECT, the actual root-finding procedure, which can stand alone;

FTNZ, the procedure defining the z formula for a Plate Case;

BFTNX, the procedure defining the X_0 formula for a Beam Case;

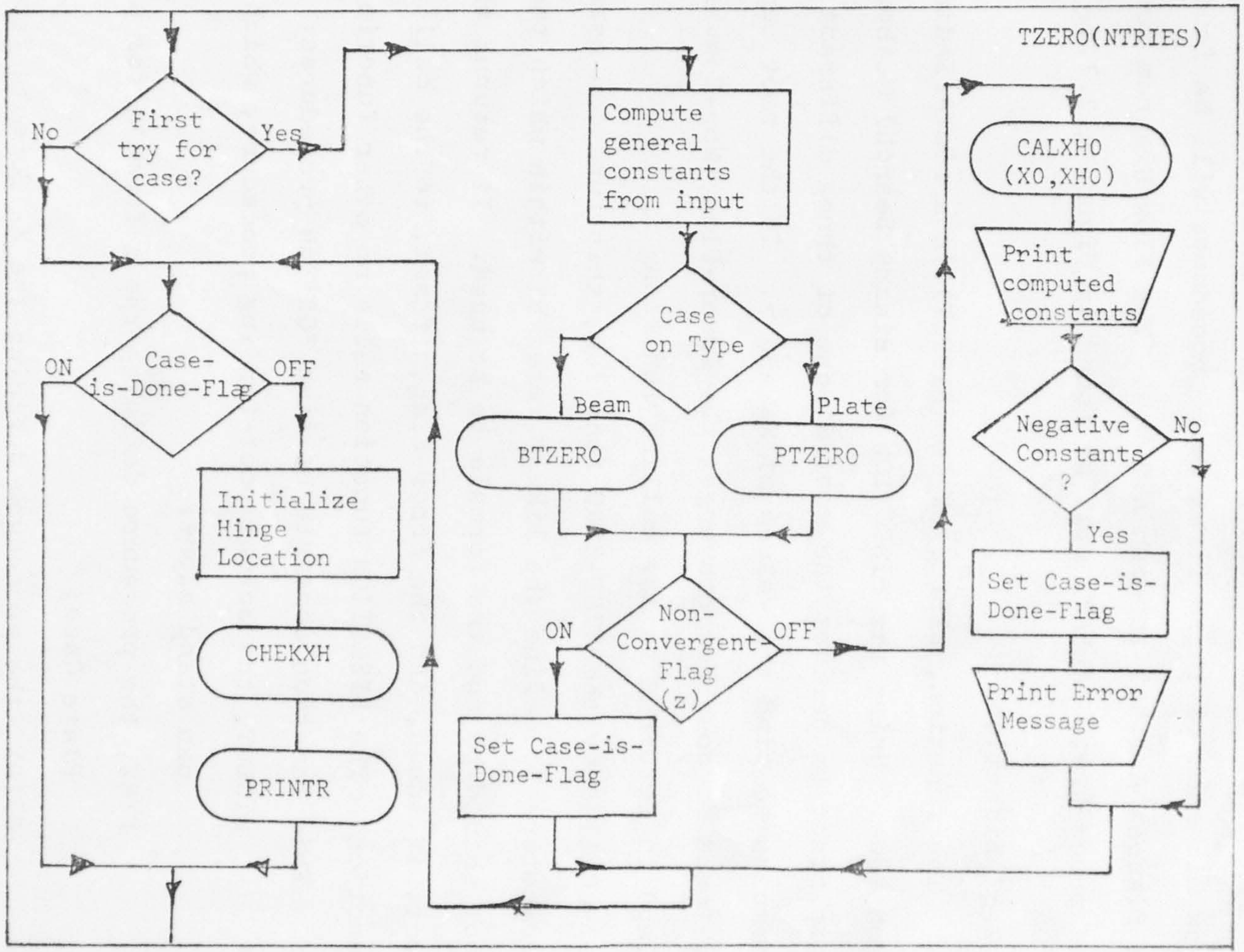


Figure A.2.4.1. Flow Diagram of Subroutine TZERO

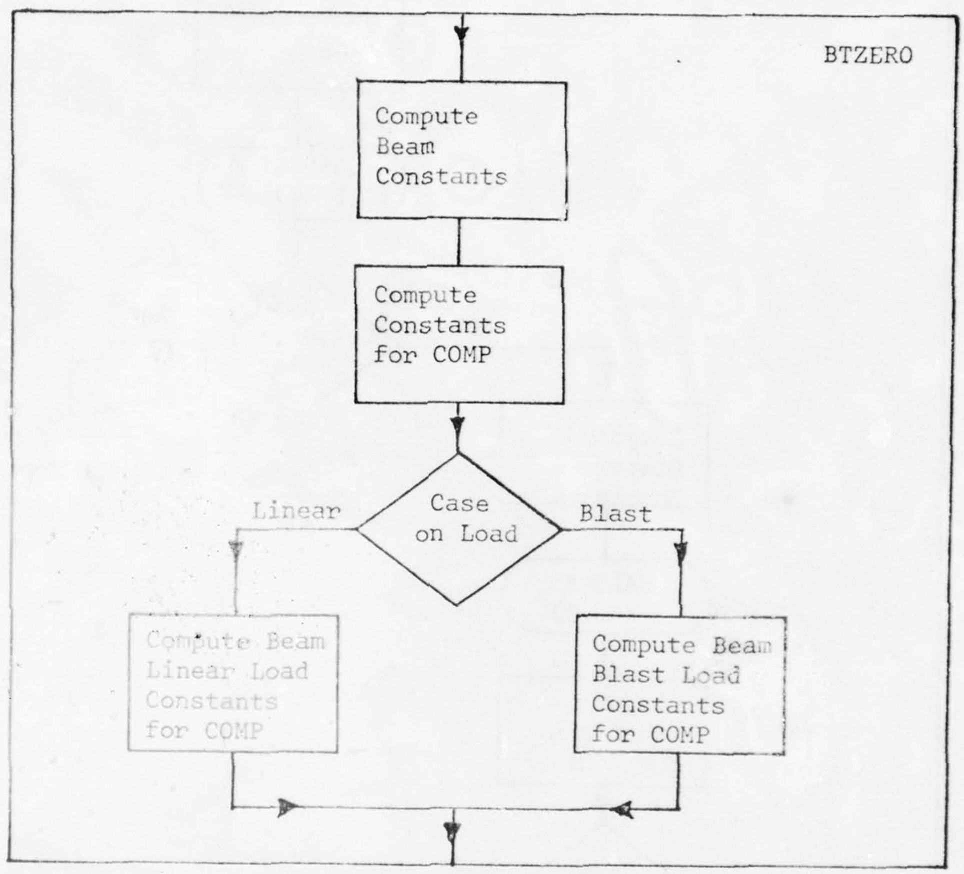


Figure A.2.4.2. Flow Diagram of Subroutine BTZERO

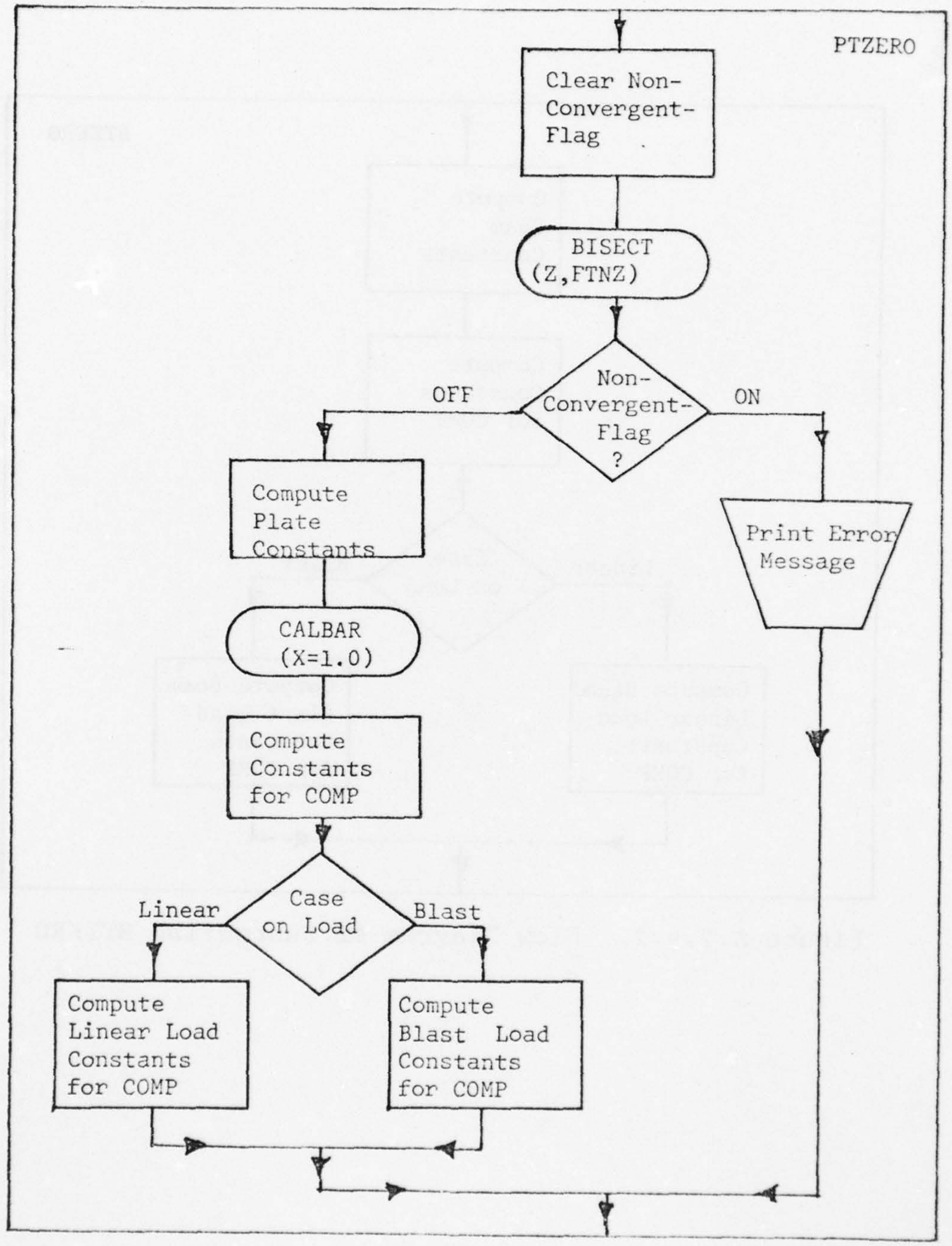


Figure A.2.4.3. Flow Diagram of Subroutine PTZERO

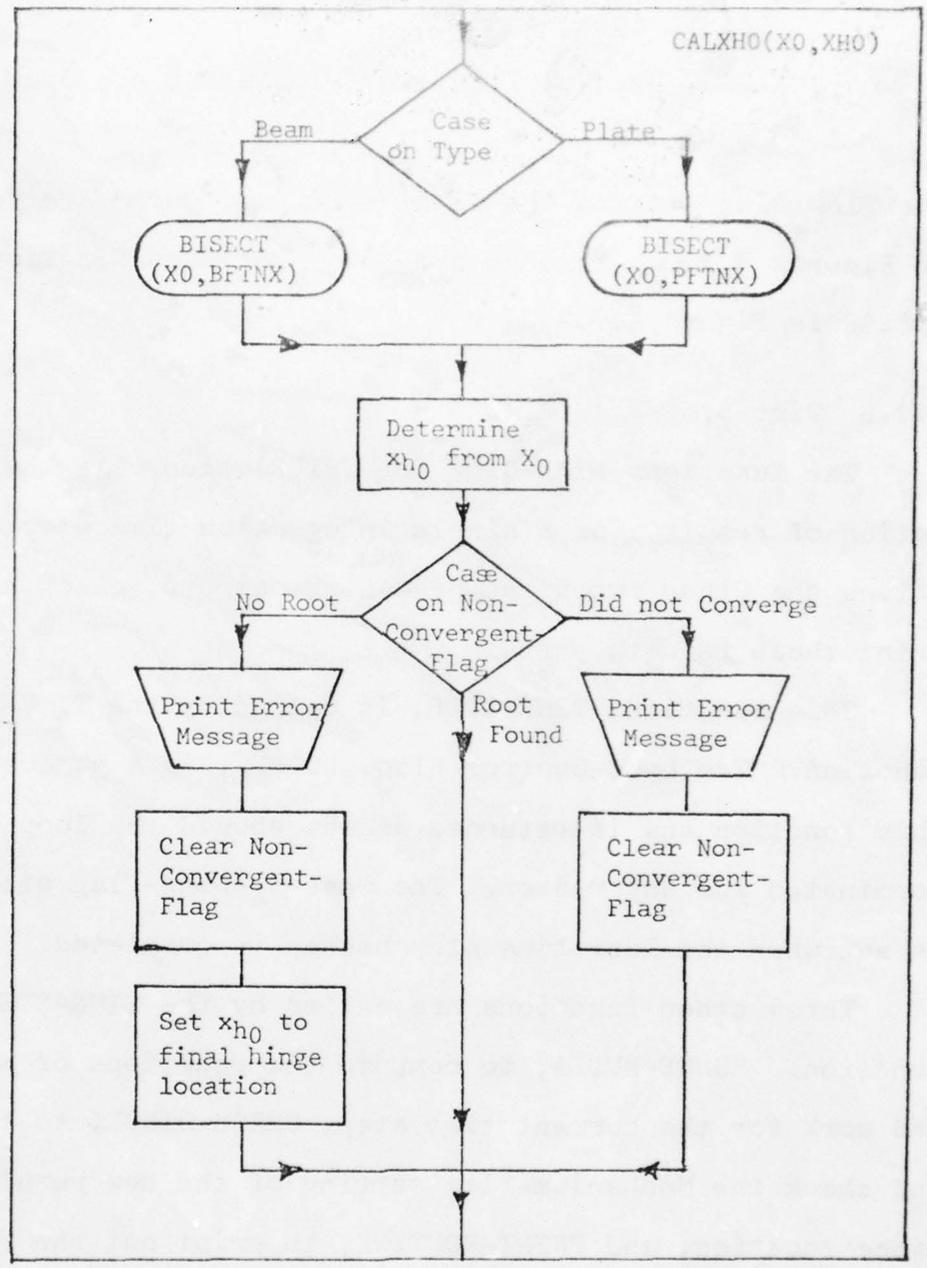


Figure A.2.4.4. Flow Diagram of Subroutine CALXHO

PFTNX, the procedure defining the X_0 formula for a Plate Case; and

CALBAR, used by PFTNX to calculate $\ddot{\theta}$ and $\ddot{\delta}$ at time zero.

The Flow Diagrams for the first four procedures are found in Figures A.2.5.1 through A.2.5.4. The Flow Diagram for CALBAR is Figure A.2.7.3.

A.2.6 Time-Step

The function, TIME-STEP (or TSTEP), controls the computation of results for a single integration time step. It drives the other functions needed to compute, check, and print these results.

This function, TIME-STEP, is called by the TIME-CONTROL function. The Loop-Control-Flag, LOOPFG, is a parameter to this function and is returned as set should the loop be terminated for any reason. The Case-is-Done-Flag will also be set when the last time step needed is completed.

Three other functions are called by the TIME-STEP function: RUNGE-KUTTA, to compute the equations of motion and work for the current time step; CHECK-HINGE, to compute and check the Mechanism-Flag setting of the new resultant hinge location; and PRINT-ROUTINE, to print out the results for this time step, if requested. The function CHECK-HINGE

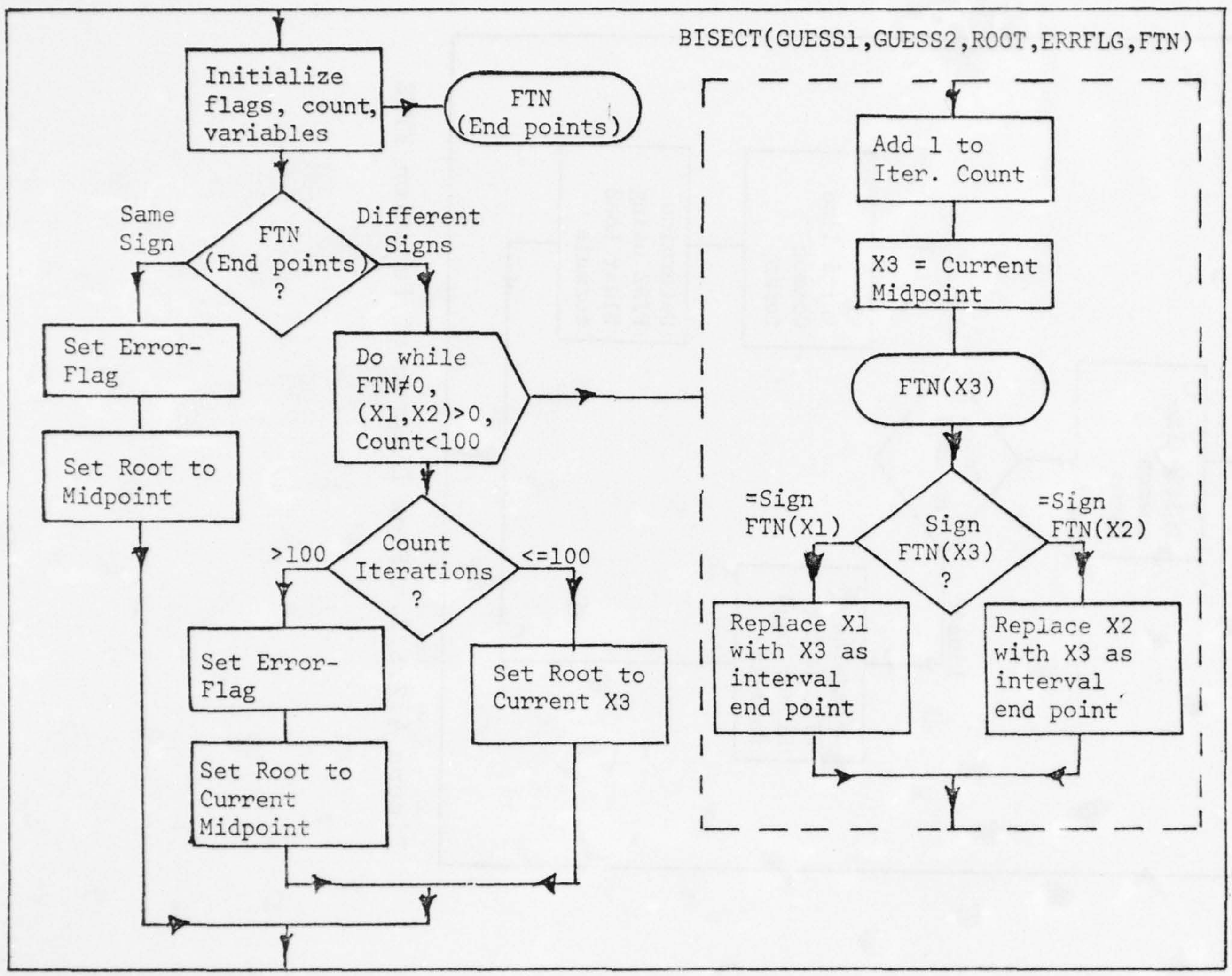


Figure A.2.5.1. Flow Diagram at Subroutine BISECT

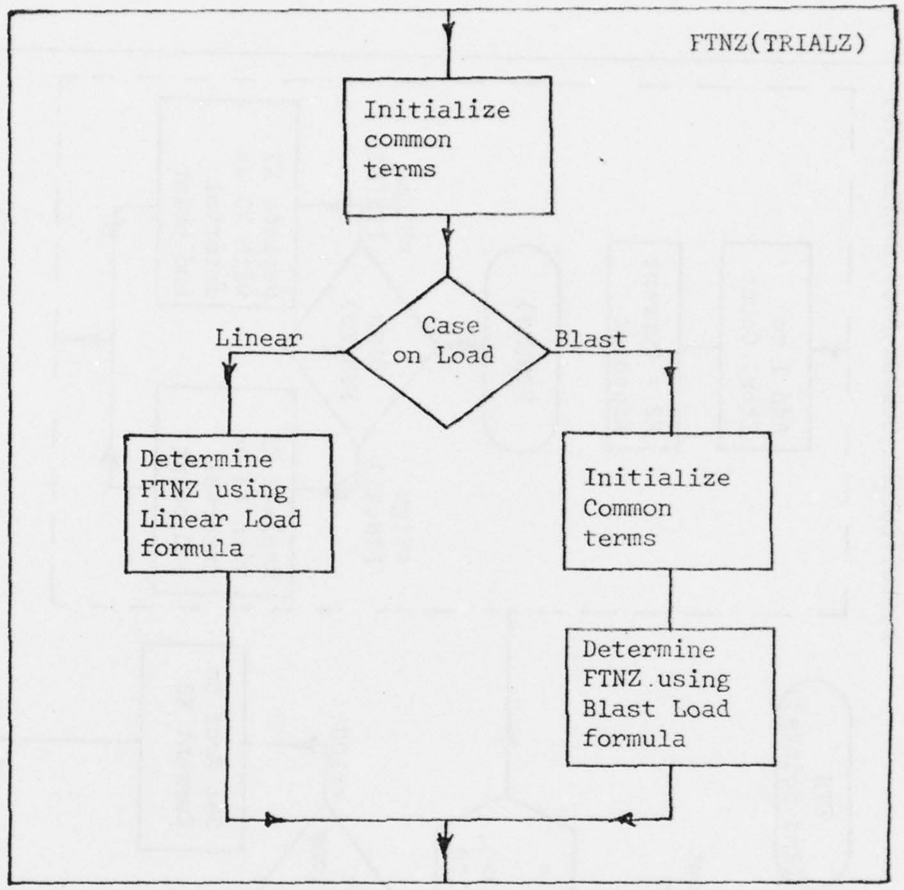


Figure A.2.5.2. Flow Diagram of Function FTNZ

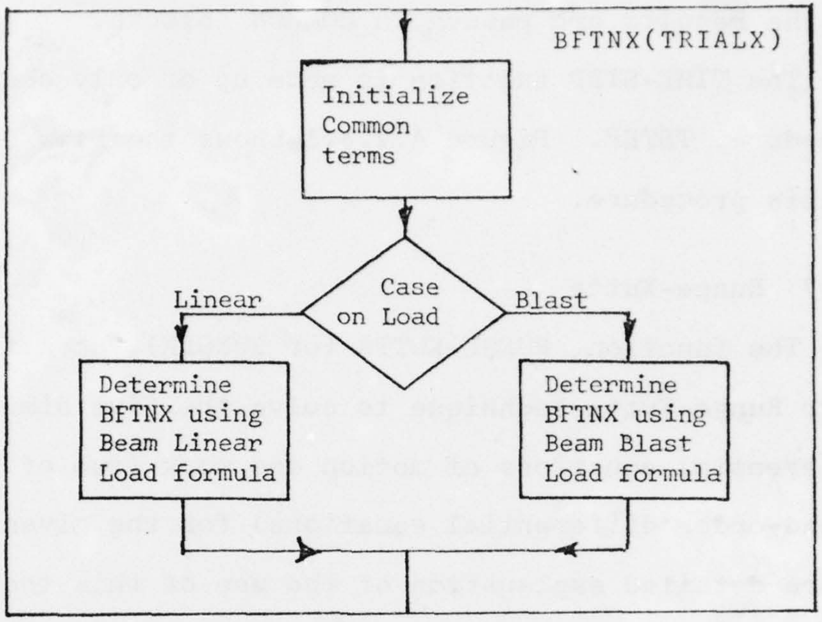


Figure A.2.5.3. Flow Diagram of Function BFTNX

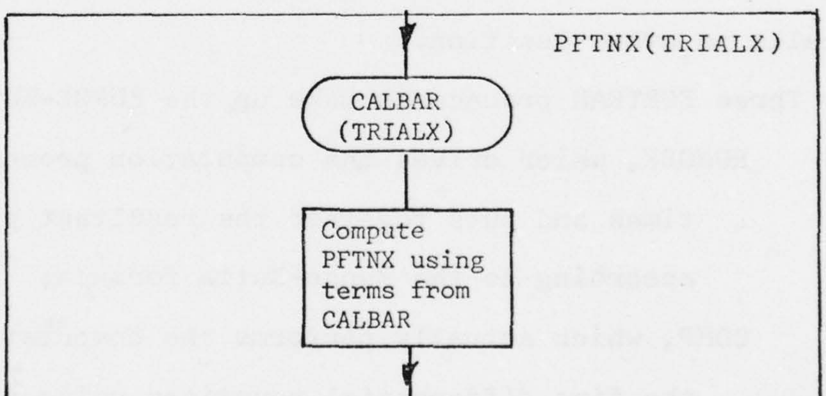


Figure A.2.5.4. Flow Diagram of Function PFTNX

will only be used while the case remains in Mechanism 2. All the results are passed in COMMON blocks.

The TIME-STEP function is made up of only one FORTRAN procedure, TSTEP. Figure A.2.6.1 shows the Flow Diagram of this procedure.

A.2.7 Runge-Kutta

The function, RUNGE-KUTTA (or RUNGEK), uses the fourth-order Runge-Kutta technique to solve the five simultaneous differential equations of motion and work (two of which are second-order differential equations) for the given time-step. A more detailed explanation of the use of this technique appears in Section A.5.2.

This function is called by the TIME-STEP function and returns computed results through the COMMON block, RESULT. It calls no other function.

Three FORTRAN procedures make up the RUNGE-KUTTA function:

- RUNGEK, which drives the computation procedure four times and puts together the resultant pieces according to the Runge-Kutta formula;
- COMP, which actually performs the computations of the five differential equations using parameters provided by RUNGEK; and
- CALBAR, which provides pieces of the $\ddot{\theta}$ and $\ddot{\delta}$ calculation for plate cases.

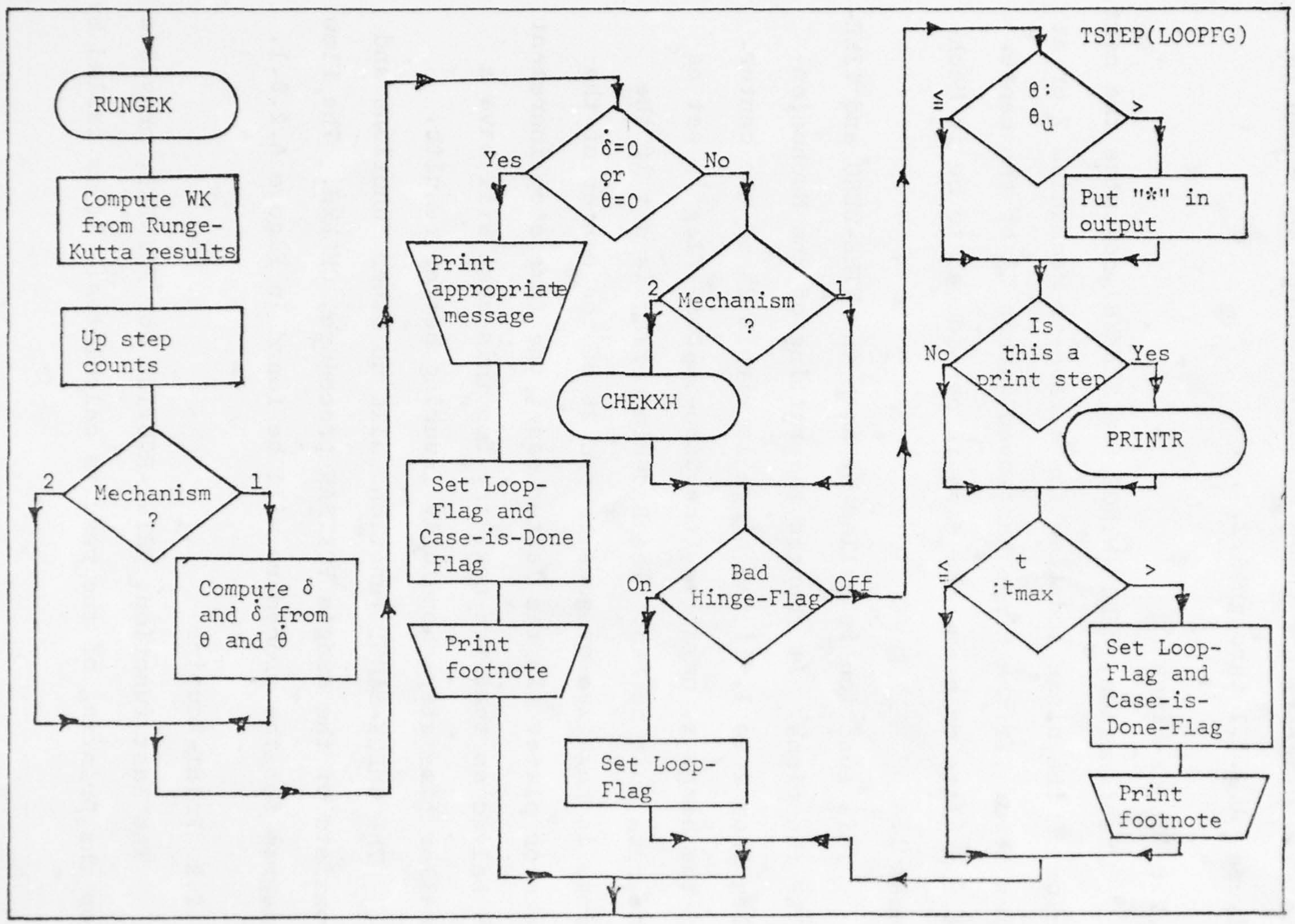


Figure A.2.6.1. Flow Diagram of Subroutine TSTEP.

The Flow Diagrams for these three procedures are shown in Figures A.2.7.1 through A.2.7.3.

A.2.8 Check-Hinge

The function, CHECK-HINGE (or CHEKXH) computes the next value of the hinge location for a case in Mechanism 2 or at time zero. If the hinge has moved within 2% of the center of the plate or beam, the case is considered to be in Mechanism 1.

This function is called by both the TIME-ZERO and TIME-STEP functions. It changes the setting of the Mechanism-Flag from 2 to 1, if the hinge is within 2% of the center. If the hinge is negative, the Case-is-Done-Flag is set as the data must be bad. The Bad-Hinge-Flag is set if the hinge is negative or goes beyond 2% of the center of the beam or plate. In the latter case, the time step increment is halved so that the next try for this case will have a smaller time step, hopefully insuring better results.

The CHECK-HINGE function calls no other functions and consists of the single FORTRAN procedure, CHEKXH. The Flow Diagram for the procedure will be found in Figure A.2.8.1.

A.2.9 Print-Routine

The last function, PRINT-ROUTINE (or PRINTR), provides for the printing of the results calculated. It is called by

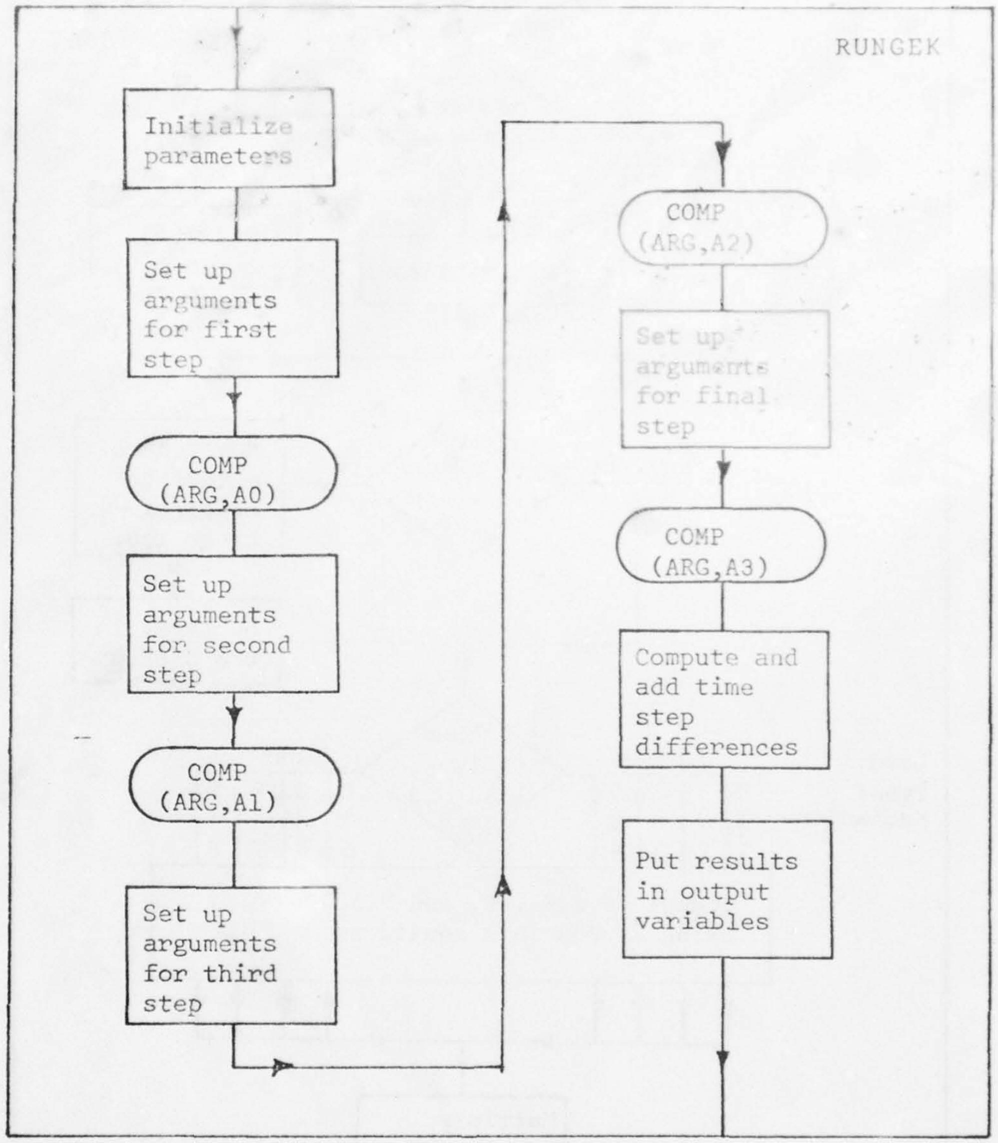


Figure A.2.7.1. Flow Diagram of Subroutine RUNGEK

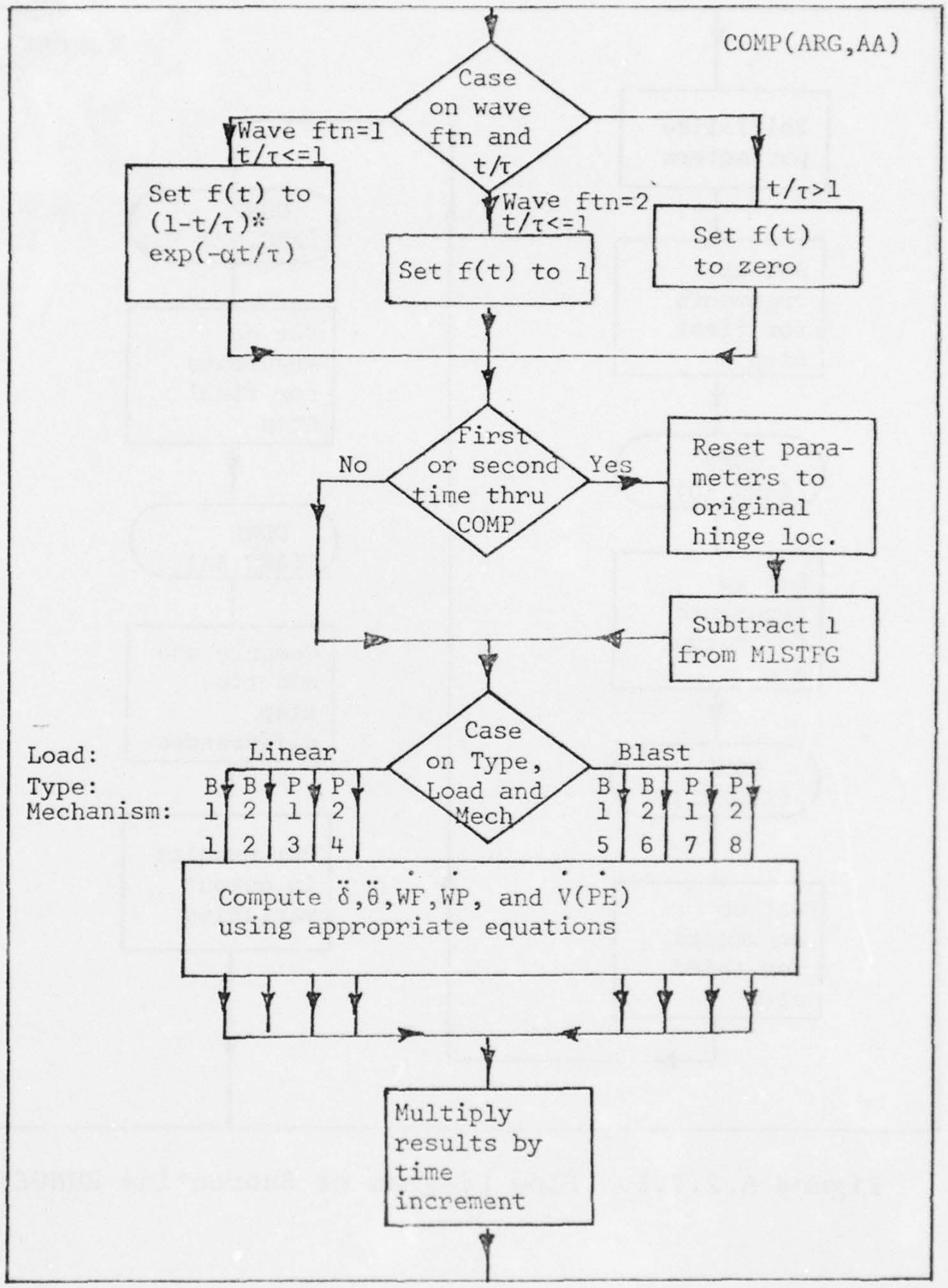


Figure A.2.7.2. Flow Diagram of Subroutine COMP

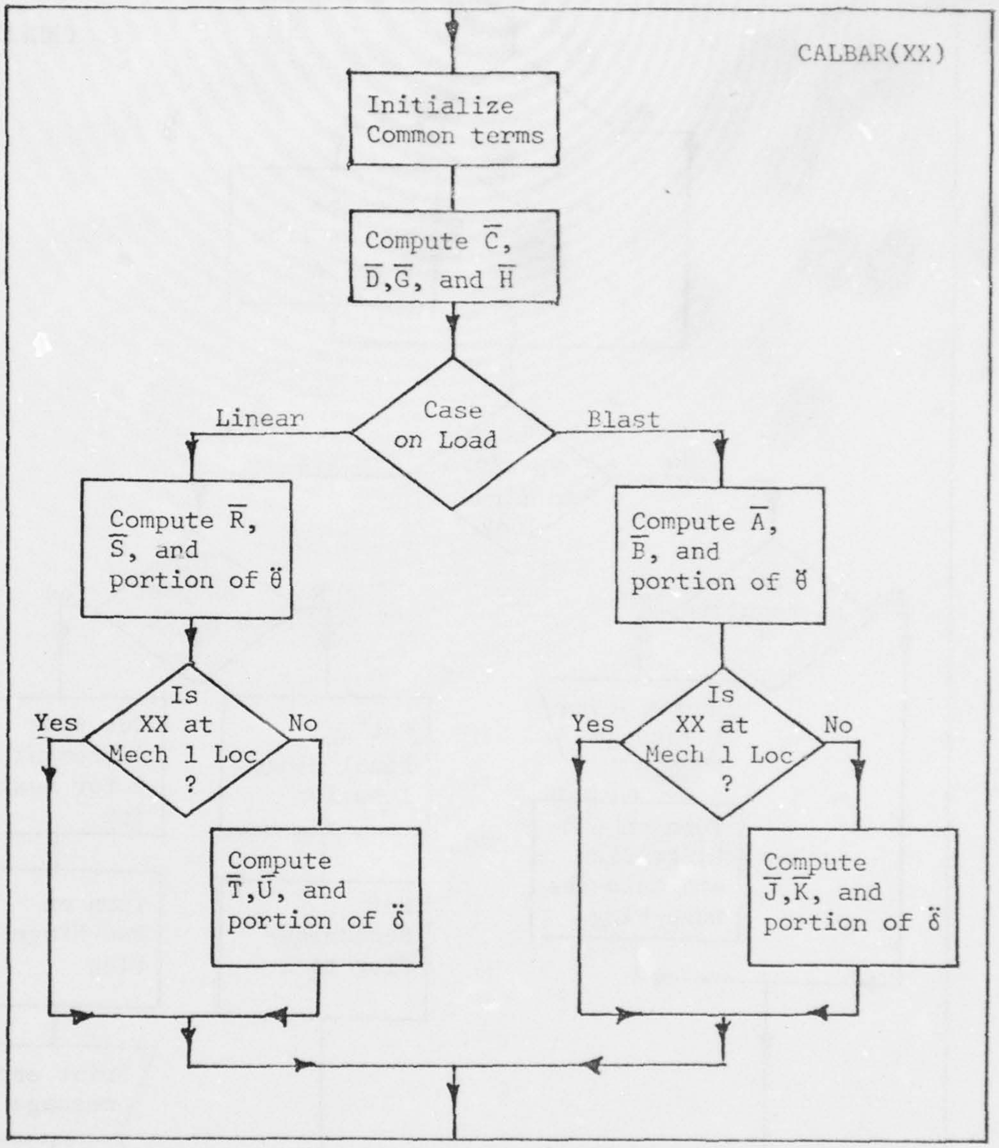


Figure A.2.7.3. Flow Diagram of Subroutine CALBAR

AD-A069 896 FLORIDA UNIV EGLIN AFB GRADUATE ENGINEERING CENTER F/G 13/13

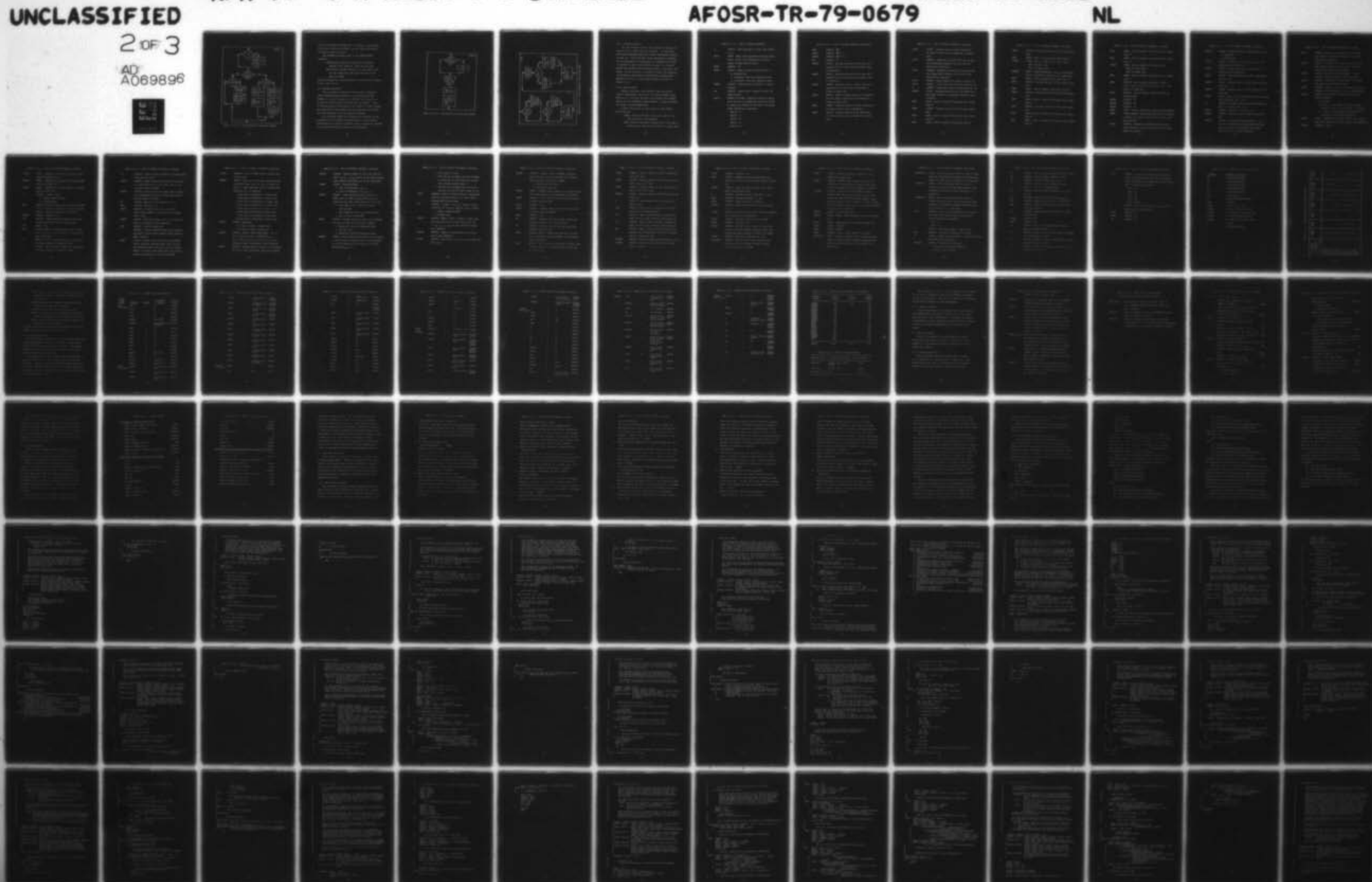
FAILURE OF UNDERGROUND HARDENED STRUCTURES SUBJECTED TO BLAST L--ETC(U)

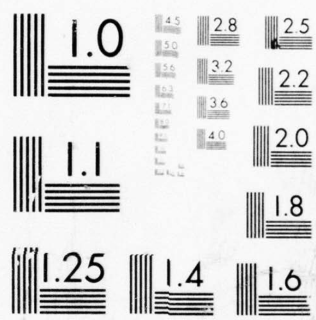
APR 79 C A ROSS, C C SCHAUBLE AFOSR-78-3592

UNCLASSIFIED AFOSR-TR-79-0679 NL

2 OF 3

AD A069896





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

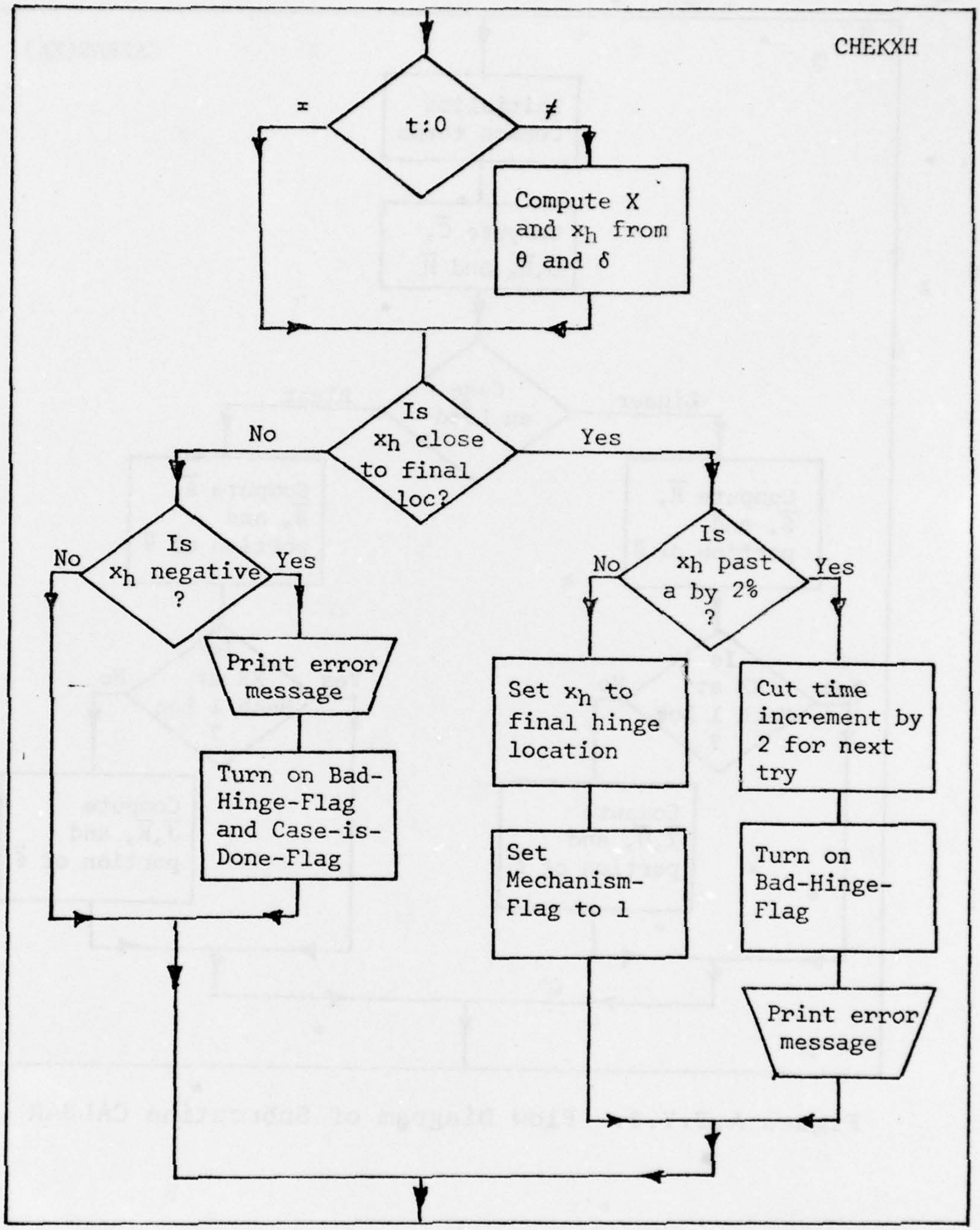


Figure A.2.8.1. Flow Diagram of Subroutine CHEKXH

both the TIME-ZERO and TIME-STEP functions, using information passed through the COMMON blocks, PRINTS and RESULT. It calls no other function.

Two FORTRAN procedures make up the PRINT-ROUTINE function:

PRINTR, which prints a single line of results, adding to the number of lines printed; and

PAGE, which provides headings on a new page of output and resets the line count whenever a new page is needed.

The Flow Diagram for these procedures can be found in Figures A.2.9.1 and A.2.9.2.

A.3 Storage Allocation

This program was written in Control Data Corporation (CDC) FORTRAN Extended, Version 4, an extension of ANSI FORTRAN IV, and was tested on the NOS/BE1 operating System of the CDC 6600 at Eglin Air Force Base in Florida. Thus, all figures which follow refer to the manner in which storage was allocated on that machine and can only serve as a model for those wishing to use the program elsewhere.

All data base tables and constants are internal to the program and are described in detail below. A more precise specification of storage allocation can be obtained by requesting a Load Map during the loading of the program.

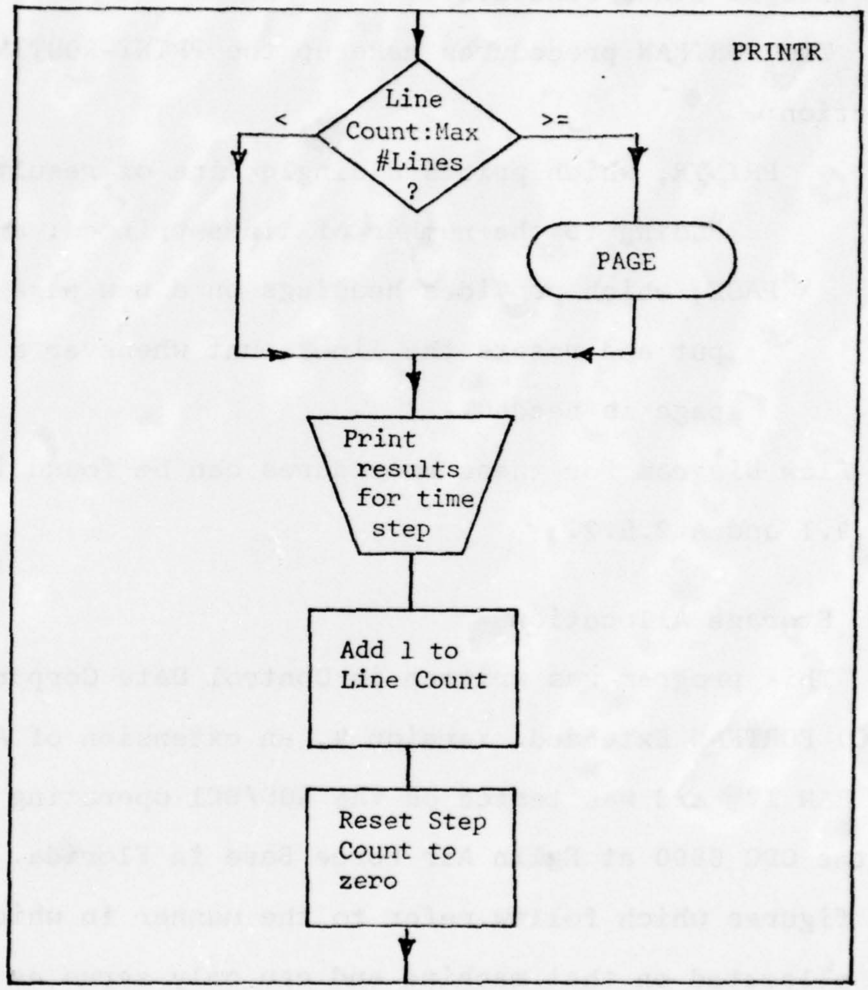


Figure A.2.9.1. Flow Diagram of Subroutine PRINTR

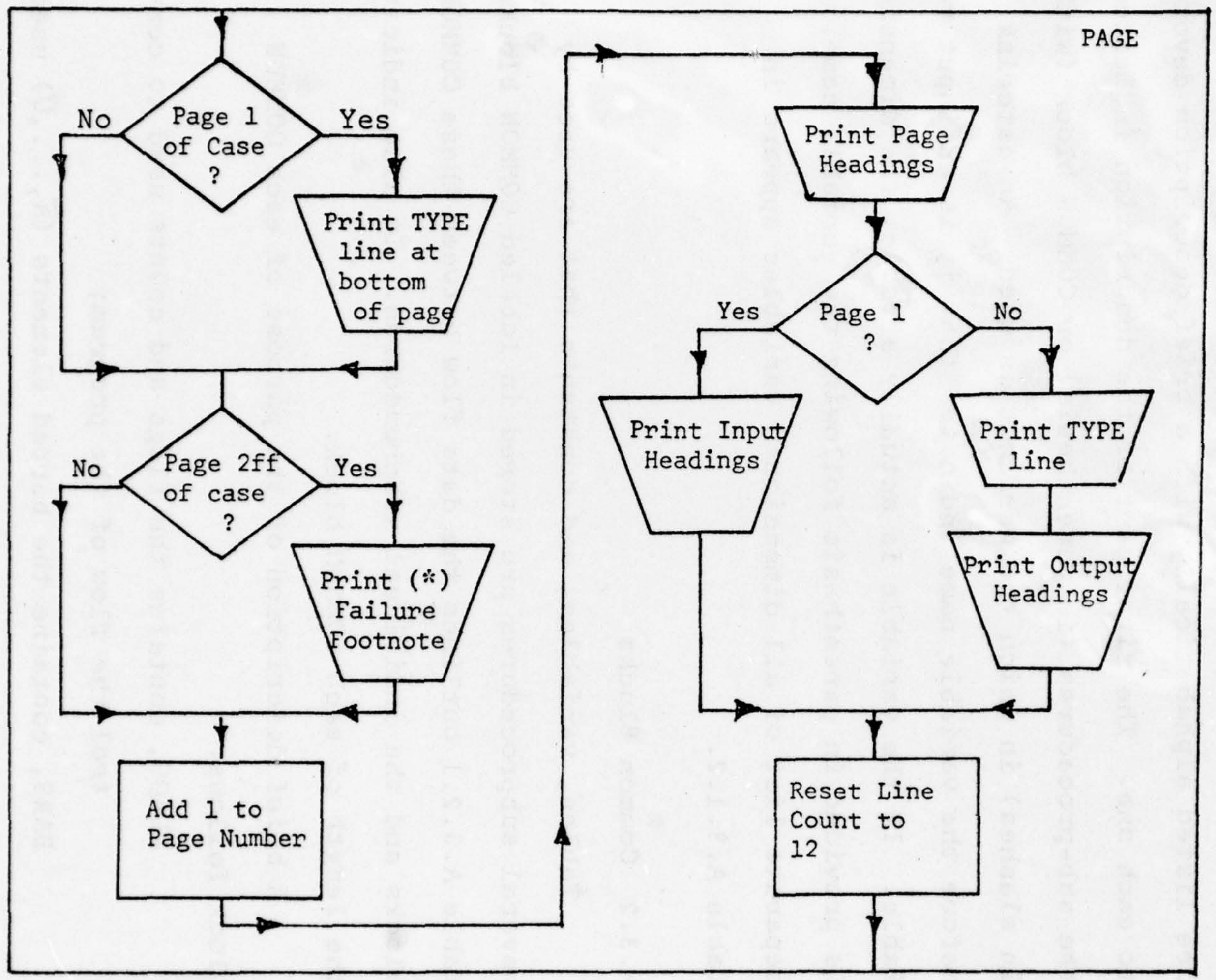


Figure A.2.9.2. Flow Diagram of Subroutine PAGE

A.3.1 Program Variables

All the variables used by the program are defined in Table A.3.1.1. This acts as a glossary. The variables are listed alphabetically with a brief description devoted to each one. The first part of the description indicates the sub-procedures (in parenthesis) or COMMON block (within slashes) in which the variable is used. An asterisk before the variable name indicates that it is an input variable. If the variable is actually a table, the dimension is provided in parenthesis following the variable name. A separate list of all dimensioned variables appears in Table A.3.1.2.

A.3.2 Common Blocks

Tables, variables, and constants that are used by several subprocedures are stored in labeled COMMON blocks. Table A.3.2.1 outlines the data flow between these COMMON blocks and the individual subprocedures. It also indicates the length of each COMMON block.

A brief description of the purpose of each COMMON block follows:

FLAGS, contains the flags and counts used to control the flow of the program;

BARS, contains the barred elements ($\bar{A},\ldots,\bar{U}$) used in computing the equations of motion of a plate case;

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES

*A	/INPUTS/ Beam Half Span or Plate Half Width, a, in.
AA(5)	(COMP) Dummy array parameter returning first and/or second order differential equation values to Runge-Kutta routine
ACUBE	/CONSTS/ a^3
*AIDA	/INPUTS/ Weight vector for plate case, n: 0, vertical wall 1, horizontal slab with explosive below -1, horizontal slab with explosive above
*ALPHA	/INPUTS/ Pressure Decay Constant, α, dimensionless
*AR	/INPUTS/ Aspect ratio, length to width, $\overline{AR}$, dimensionless
ARG(5)	(RUNGEK and COMP) Arguments passed by Runge-Kutta routine to computation routine in order to obtain new values for the first and second order differential equations: ARG(1) = t ARG(2) = δ ARG(3) = $\dot{\delta}$ ARG(4) = θ ARG(5) = $\dot{\theta}$

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

ARSQ	/CONSTS/	$\overline{AR}^2$
ARSZP1	/CONSTS/	$\overline{AR}^2 Z + 1$
ARZSP1	/CONSTS/	$\overline{AR}\ Z^2 + 1$
ASQ	/CONSTS/	a^2
ATHEDB	/COMPS/	Constant portion of first part of $\ddot{\theta}$ computation for a Beam Case with Blast load in Mechanism 1
ATHEDL	/COMPS/	Constant portion of first part of $\ddot{\theta}$ computation for a Beam Case with Linear load in Mechanism 1
ATHED1	/COMPS/	Constant portion of first part of $\ddot{\theta}$ computation for Plate case in Mechanism 1, Linear or Blast load
ATHED2	/COMPS/	Second constant portion of $\ddot{\theta}$ computation for Beam or Plate case in Mechanism 1, Linear or Blast load
AVDOT	/COMPS/	Constant portion of $\dot{V}(\dot{PE})$ computation for Beam or Plate case in Mechanism 1
AWFDB	/COMPS/	Constant portion of $\dot{W}_f$ computation for Beam or Plate Case in Mechanism 1, Blast load

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

AWFDL	/COMPS/	Constant portion of $\dot{W}F$ computation for Beam or Plate case in Mechanism 1, Linear load
A0(5)	(RUNGEK)	Computation results for first of the four Runge-Kutta formulas, a_0, for all five differential equations
A1(5)	(RUNGEK)	Computation results for second of the four Runge-Kutta formulas, a_1, for all five differential equations
A2(5)	(RUNGEK)	Third Runge-Kutta formula results, a_2
A3(5)	(RUNGEK)	Fourth Runge-Kutta formula results, a_3
B	/CONSTS/	Plate Half Length or Beam Width, b, in.
BADXFG	/FLAGS/	Bad-Hinge-Flag set to 1 by CHEKXH if a bad hinge location is computed; otherwise, zero (Integer)
BARA	/BARS/	Used to compute $\ddot{\theta}$ for plate case, blast load, $\overline{A}$
BARA1	/CONSTS/	Constant portion of $\overline{A}$ for given plate case
BARB	/BARS/	Used to compute $\ddot{\theta}$ for plate case, blast load, $\overline{B}$
BARC	/BARS/	Used to compute $\ddot{\theta}$ for plate case, $\overline{C}$

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

BARC1	/CONSTS/	Constant portion of $\overline{C}$ for given plate case
BARD	/BARS/	Used to compute $\ddot{\theta}$ for plate case, $\overline{D}$
BARDEL	/BARS/	Used to compute $\ddot{\delta}$ for plate case: $\overline{J}*\overline{K}$, if blast load; $\overline{T}*\overline{U}$, if linear load
BARDNM	/BARS/	Common denominator used in calculating $\overline{A}$, $\overline{D}$, $\overline{H}$, and $\overline{R}$ for a plate case $\ddot{\theta}$ computation
BARG	/BARS/	Used to compute $\ddot{\theta}$ for plate case, $\overline{G}$
BARG1	/CONSTS/	Constant portion of $\overline{G}$ for given plate case
BARH	/BARS/	Used to compute $\ddot{\theta}$ for plate case, $\overline{H}$
BARH1	/CONSTS/	Constant portion of $\overline{H}$ for given plate case
BARJ	/BARS/	Used to compute $\ddot{\delta}$ for plate case, blast load, $\overline{J}$
BARJ1	/CONSTS/	Constant portion of $\overline{J}$ for given plate case
BARK	/BARS/	Used to compute $\ddot{\delta}$ for plate case, blast load, $\overline{K}$
BARR	/BARS/	Used to compute $\ddot{\theta}$ for plate case, linear load, $\overline{R}$

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

BARS	/BARS/ Used to compute $\ddot{\theta}$ for plate case, linear load, $\overline{S}$
BART	/BARS/ Used to compute $\ddot{\delta}$ for plate case, linear load, $\overline{T}$
BARTHE	/BARS/ Used to compute $\ddot{\theta}$ for plate case: $\overline{A}*\overline{B}$, if blast load $\overline{R}*\overline{S}$, if linear load
BARU	/BARS/ Used to compute $\ddot{\delta}$ for plate case, linear load, $\overline{U}$
BEAM	(GETITL) Five constant characters, "BEAM ", used to match first five columns of first input cards as a Beam case
*BETA	/INPUTS/ Spatial Pressure Decay Constant, β, dimensionless
BETACB	/CONSTS/ β^3
BETAFR	/CONSTS/ β^4
BETASQ	/CONSTS/ β^2
BETAX	(COMP) Used to consolidate calculations, β*CURRX
BFTNX	(BFTNX,CALXH0) Subfunction and value of function defining X_0, used by BISECT in determining root for Beam case
BLANK	(TCNTRL) Alphabetic constant containing one blank character, used to initialize the Failure Flag for printout

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

BPFLAG	/FLAGS/	Beam-Plate-Flag to denote type of case: 1, Beam Case; 2, Plate Case
BTHED2	/COMPS/	Constant portion of second part of $\ddot{\theta}$ computation for Beam Case
BTHED3	/COMPS/	Constant portion of third part of $\ddot{\theta}$ computation for Beam Case
BVDOT	/COMPS/	Constant portion of $\dot{V}(\dot{P}E)$ computation for Beam Case
BWFDOT	/COMPS/	Constant portion of $\dot{W}F$ computation for Beam Case
BWPDOT	/COMPS/	Constant portion of $\dot{W}P$ computation for Beam Case
CURRX	(COMP)	Value of X as determined by current Runge-Kutta arguments for δ and θ.
*D	/INPUTS/	Distance between reinforcing rod and edge of opposite side, d, in.
DELTA	/RESULT/	Midpoint deflection, δ, in.
DELTAK	/CONSTS/	Constant used by several equations, nw/m
DONEFG	/FLAGS/	Case-is-Done-Flag (integer): Set to one when a condition is found which indicates the case should be terminated (including a normal end); set to zero otherwise

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

DX(5)	(RUNGEK) Runge-Kutta computed first differential changes for given time step for the five differential equations
DXDX(2)	(RUNGEK) Runge-Kutta computed second differential changes for given time step for the two second order differential equations
EOFLAG	/FLAGS/ End-of-File-Flag (integer): Set to one when input is expended; zero otherwise
EPSILN	(BISECT) Acceptable accuracy error in convergence used by root-finding routine
EPSLNU	/CONSTS/ Ultimate strain of reinforcing element, assumed to be a constant, 0.2, ε_u, dimensionless
ERRFLG	(BISECT) Error-Flag returned to calling procedure (integer): 0 = no error +1 = error: even number of roots in given interval, could not use method to find root -1 = error: method would not converge after 100 tries, root may be acceptable
EXPBCX	(COMP) $\exp\beta$(CURRX-1), where CURRX is the current value of X being used in the Runge-Kutta computation
EXPBET	/CONSTS/ $\exp(-\beta)$

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

EXPBX1	/BARS/	$\exp\beta(XX-1)$, where XX is current value being considered for X
EXPBZX	/BARS/	$\exp\beta(Z*XX-1)$, where XX is current value being considered for X
EXPBZ1	(FTNZ)	$\exp\beta(TRIALZ-1)$, where TRIALZ is current guess at value for z
*F	/INPUTS/	Support factor: 1 = simply supported edges 2 = clamped edges
FLAG	/PRINTS/	Failure Flag, used to indicate whether or not reinforcing element has fractured (alphabetic): blank, no fracture; *, fracture
FOURTH	/CONSTS/	Constant, 1/4
FTN	(BISECT)	Dummy parameter naming function subprocedure for which root is being found
FTNT	(COMP)	f(t)
FTNZ	(PTZERO, FTNZ)	Subfunction and value of function defining z, used by BISECT in determining root, plate case
F1	(BISECT)	Value of given function at lower end of current interval of guesses, f(X1)
F2	(BISECT)	Value of given function at upper end of current interval of guesses, f(X2)

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

F3	(BISECT) Value of given function at current guess being tested, f(X3)
GUESS1	(BISECT) Dummy variable for lower end of interval containing root used as first seed for Bisection method in BISECT
GUESS2	(BISECT) Dummy variable for upper end of interval containing root used as second seed for Bisection method in BISECT
*H	/INPUTS/ Beam or Plate thickness, h, in.
HALF	/CONSTS/ Constant, 1/2
HALFTI	(RUNGEK) One-half given time step, dt/2
I	(COMP, PAGE, RUNGEK) Array subscript varying over a loop
IERRFG	/FLAGS/ Input-Error-Flag: Set to one when bad data is encountered; zero otherwise
ITER	(BISECT) Current interation
J	(CONCRE) Current Computer System print density setting as returned from System Library routine, PDEN, lines/in.
KOUNT	/FLAGS/ Current total for run of accumulated errors caused by improper input data, used by CDC FORTRAN Extended Version 4 library routine ERRSET, to inhibit job termination for bad data; maximum arbitrarily set to 100 in CONCRE

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

LCOUNT	(RWDATA)	Value of KOUNT before reading data for new case
LMBPSM	(COMP)	Coded sum of the Load, Mechanism, and Beam-Plate-Flags plus one, used to determine which of eight different sets of differential equations will be computed: 1 Beam Case in Mechanism 1, Linear Load 2 Beam Case in Mechanism 2, Linear Load 3 Plate Case in Mechanism 1, Linear Load 4 Plate Case in Mechanism 2, Linear Load 5 Beam Case in Mechanism 1, Blast Load 6 Beam Case in Mechanism 2, Blast Load 7 Plate Case in Mechanism 1, Blast Load 8 Plate Case in Mechanism 2, Blast Load
LOADFG	/FLAGS/	Load-Flag: 1, if case in under a Linear Load 2, if case in under a Blast Load
LOOPFG	(TCNTRL, TSTEP)	Loop-Control-Flag: Set to one if loop for time = t_1(TINCR),t_2,..., t_{max} (TMAX) should be exited for any reason, including normal termination; zero otherwise
MAXLIN	/PRINTS/	Maximum number of lines to be printed per output page; determined from print density

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

MAXTRI	(TCNTRL) Maximum number of times the loop for time = $t_1, t_2, \ldots, t_{max}$ can be tried using successively smaller time steps, if exited abnormally	
MECHFG	/FLAGS/ Mechanism-Flag:	
	1	Case in Mechanism 1, $x_h = a$
	2	Case in Mechanism 2, $0 \leq x_h < a$
M1STFG	/FLAGS/ First-Time-Flag used to provide a dummy value of CURRX to prime the first two passes through COMP during the first time step:	
	0	Have completed at least two passes through COMP
	1	Have completed first pass through COMP
	2	Have not yet called COMP
NAIDA	/PRINTS/ Integer value of n plus two; subscript for array of type of slab titles, TYPSLB	
	1	Horizontal Slab with Explosive above
	2	Vertical Wall
	3	Horizontal Slab with Explosive below
NCONFG	/FLAGS/ Non-Convergent-Root-Flag used to signify if root-finding technique found a non-convergent or out-of-bounds root for the given function determining x_{h_0} or z:	

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

	0 Root found is valid
	1 Root did not converge, was out-of-range, or could not be found with this method using seeds (guesses) given
	-1 Root did not converge after 100 interactions, but may be close enough to use
NERR	(RWDATA) Count of errors found by reading bad data for the given case; i.e., input errors accumulated for given case
NF	/PRINTS/ Integer value of F, Support Factor; subscript for array of support titles, TYPSUP: 1 Simply supported edges; 2 Clamped edges
NTRIES	(DRIVER, TCNTRL, TZERO) Number of times loop for computing and printing output lines for $t = t_1, t_2, \ldots, t_{max}$ has been tried for given case ($\leq$MAXTRI)
NUMLIN	/PRINTS/ Number of lines printed on current output page ($\leq$MAXLIN)
NUMPAG	/PRINTS/ Page number of current output page for given case

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

NWAVEF	/PRINTS/	Integer value of WAVEFN, Pressure Wave Function; used as subscript for Pressure Wave Function Title array, TYPWAV: 1 General time function 2 Square Wave Function
ONEMCX	(COMP)	1-CURRX, where CURRX is the current value of X being used by the Runge-Kutta computation
ONEMXX	(CALBAR)	1-XX, where XX is the current value of X being used by CALBAR
ONEMZ	/CONSTS/	1-z
ONEMZX	(CALBAR)	1-z*XX, where XX is the current value of X being used by CALBAR
ONEPZ	/CONSTS/	1+z
*PC	/INPUTS/	Pressure at plate or beam center for varying portion of the load, P_C, psi
*PE	/INPUTS/	Pressure at plate or beam edges for constant load, P_E, psi
PFTNX	(CALXHO, PFTNX)	Subfunction and value of function defining x_{h_0} used by BISECT in determining root, plate case
PLATE	(GETITL)	Five constant characters, "PLATE", used to match first five columns of first input card as a Plate case

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

PVDOT	/COMPS/	Constant portion of $\dot{V}(\dot{PE})$ computation for plate case
PWFDB	/COMPS/	Constant portion of $\dot{WF}$ computation for plate case, blast load
PWFDL	/COMPS/	Constant portion of $\dot{WF}$ computation for plate case, linear load
PWPDOT	/COMPS/	Constant portion of $\dot{WP}$ computation for plate case
Q	/INPUTS/	Reinforcement Ratio in Tension, q, dimensionless
ROOT	(BISECT)	Root found by Bisection Method and returned to calling procedure
SF1	(BISECT)	Sign of given function evaluated at lower end of current interval containing root
SF2	(BISECT)	Sign of given function evaluated at upper end of current interval containing root
SF3	(BISECT)	Sign of given function evaluated at midpoint (or current guess at root) of current interval containing root
*SIGMAC	/INPUTS/	Concrete compressive strength, σ_c, psi
*SIGMAR	/INPUTS/	Reinforcing element yield/ultimate stress, σ_r, psi

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

SIXTH	/CONSTS/	Constant, 1/6
STAR	(TSTEP)	Alphabetic constant containing one asterisk, used to set the Failure Flag for printout
STEPCT	/PRINTS/	Count of number of time steps taken since last printout line
T	/RESULT/	Current time for time step, t, sec.
*TAU	/INPUTS/	Pressure duration, τ, sec.
THETA	/RESULT/	Hinge rotation, θ, radians
THETAD	/RESULT/	First differential of θ with respect to t, $\dot{\theta}$
THETAU	(TSTEP)	Rotation for ultimate strain of reinforcing element, θ_u, dimensionless
THETU1	/CONSTS/	Constant portion of θ_u, $4h\varepsilon_u/d^2$
THIRD	/CONSTS/	Constant, 1/3
TIMNOW	/PRINTS/	Current time at start of run as returned from System Library routine, TIME, used in printing page headings to define run output
*TINCR	/INPUTS/	Time step increment, dt, sec.
*TITLE(15)	/PRINTS/	Title of given case provided from case header card and used in page headings; 15 x 5 characters in length

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

*TMAX	/INPUTS/	Maximum allowable value for t for given case, t_{max}, sec.
TODAY	/PRINTS/	Current date at start of run as returned from System Library routine, DATE: used in printing page headings to define run output
*TPRINT	/PRINTS/	Interval of time allowed between printed time steps, t_{print}, sec.: $t_{print} \leq dt$, every time step will be printed; $t_{print} > t_{max}$, no time steps will be printed; $dt < t_{print} \leq t_{max}$, only those times steps which are rounded multiples of t_{print} will be printed
TRATIO	(COMP)	t/τ, dimensionless
TRIALX	(BFTNX, PFTNX)	Dummy parameter for current guess at X
TRIALZ	(FTNZ)	Dummy parameter for current guess at z
TRXSQ	(BFTNX)	$TRIALX^2$
TRZSQ	(FTNZ)	$TRIALZ^2$
*TYPE	/PRINTS/	Type of Case, "BEAM " or "PLATE"
TYPLOA(2)	(PAGE)	Array containing title for Type of Load, used for printing page headings; subscripted by LOADFG (Load-Flag); each title contains 1 X 6 characters

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

TYPSLB(4,3)	(PAGE)	Array containing title for Type of Slab, used for printing page headings; subscripted by NAIDA (n); each title contains 4 X 8 characters
TYPSUP(2)	(PAGE)	Array containing title for Type of Support, used for printing page headings; subscripted by NF(F); each title contains 1 X 7 characters.
TYPWAV(2,2)	(PAGE)	Array containing title for Type of Pressure Wave Function, used for printing page headings; subscripted by NWAVEF (WAVEFN); each title contains 2 X 6 characters
U(2)	(RUNGEK)	Value of the variables δ and θ after the Runge-Kutta method solves their second-order differential equations: $U(1) = \delta$ $U(2) = \theta$
V	/RESULT/	Potential energy, V(PE), in-lbs.
VEL	/RESULT/	Mid-point velocity, $\dot{\delta}$, in./sec.
W	/CONSTS/	Weight per unit area, w = mg, lbf./in.^{2}, g gravitational acceleration
*WAVEFN	/INPUTS/	Pressure Wave Function: 1 General Pressure Wave Function; 2 Square Wave Function

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Continued

WF	/RESULT/	Work done by Pressure Load, WF, in-lbs.
WK	/RESULT/	Kinetic energy, WK(KE), in-lbs.
WP	/RESULT/	Plastic energy, WP, in-lbs.
X	/RESULT/	Ratio of current hinge location, x_h/a $(0 < X \leq 1)$, X, dimensionless
XH	/RESULT/	Current hinge location $(0 \leq x_h \leq a)$, x_h, in.
XH0	(CALXH0)	Original hinge location, x_{h_0}, in.
*XM	/INPUTS/	Mass per unit area, m, lbf-sec^2/in^3
XMU	/CONSTS/	Hinge moment per unit length, plate case, M_u, lb-in/in.
XX	(CALBAR)	Dummy parameter for value of X used by subroutine
XXCUBE	(CALBAR)	XX^3
XXSQ	(CALBAR)	XX^2
X0	(CALXH0)	Ratio of original hinge location, x_{h_0}/a, X_0, dimensionless
X1	(BISECT)	Value of lower end of interval containing current guess at root
X2	(BISECT)	Value of upper end of interval containing current guess at root
X3	(BISECT)	Value of midpoint of interval; i.e., current guess at root

TABLE A.3.1.1. LIST OF PROGRAM VARIABLES, Concluded

Y(5)	(RUNGEK)	Value of the first differentials, $\dot{\delta}$ and $\dot{\theta}$, and the variables, WF, WP, and V(PE), after the Runge-Kutta method solves their first order differential equations: Y(1) = $\dot{\delta}$ Y(2) = $\dot{\theta}$ Y(3) = WF Y(4) = WP Y(5) = V(PE)
Z	/CONSTS/	Ratio of final hinge location to b, plate case, z, dimensionless
ZCUBE	/CONSTS/	z^3
ZFOUR	/CONSTS/	z^4
ZSQ	/CONSTS/	z^2

TABLE A.3.1.2. DIMENSIONS OF CONCRE VARIABLES

VARIABLE	DIMENSION (Contents)
AA	5 ($\ddot{\delta}, \ddot{\theta}, \dot{WF}, \dot{WP}, \dot{V}(\dot{PE})$)
A0	5 ($\ddot{\delta}, \ddot{\theta}, \dot{WF}, \dot{WP}, \dot{V}(\dot{PE})$)
A1	5 ($\ddot{\delta}, \ddot{\theta}, \dot{WF}, \dot{WP}, \dot{V}(\dot{PE})$)
A2	5 ($\ddot{\delta}, \ddot{\theta}, \dot{WF}, \dot{WP}, \dot{V}(\dot{PE})$)
A3	5 ($\ddot{\delta}, \ddot{\theta}, \dot{WF}, \dot{WP}, \dot{V}(\dot{PE})$)
ARG	5 ($dt, \delta, \dot{\delta}, \theta, \dot{\theta}$)
DX	5 ($d\dot{\delta}, d\dot{\theta}, dWF, dWP, dV(dPE)$)
DXDX	2 ($d\delta, d\theta$)
TITLE	15 (15A5, Case Title)
TYPLOA	2 (2A6, Descriptive Title)
TYPSLB	4,3 (3(4A8), Descriptive Title)
TYPSUP	2 (2A7, Descriptive Title)
TYPWAV	2,2 (2(2A6), Descriptive Title)
U	2 (δ, θ)
Y	5 ($\dot{\delta}, \dot{\theta}, WF, WP, V(PE)$)

TABLE A.3.2.1. CONCRE PROCEDURES AND COMMON BLOCKS

COMMON Block (Length) / Procedure Name	FLAGS (10)	BARS (17)	COMPS (16)	CONSTS (30)	INPUTS (19)	PRINTS (26)	RESULT (11)
CONCRE	X			X		X	
DRIVER	X						
NXTCAS	X					X	
GETITL	X					X	
RWDATA	X				X	X	
TCNTRL	X				X	X	X
TZERO	X			X	X		X
BTZERO	X		X	X	X		
PTZERO	X	X	X	X	X		
CALXHO	X				X		
BISECT							
FTNZ	X			X	X		
BFTNX	X			X	X		
PFTNX		X		X	X		
TSTEP	X			X	X	X	X
RUNGEK	X				X		X
COMP	X	X	X	X	X		X
CALBAR	X	X		X	X		
CHEKXH	X				X		X
PRINTR						X	X
PAGE	X					X	

COMPS, contains constants and variables used by the subroutine COMP to compute the equations of work and motion;

CONSTS, contains miscellaneous constants and variables used throughout the program;

INPUTS, contains the input data for the case;

PRINTS, contains counts and alphabetic information used for printing the output headings, etc; and

RESULT, contains the numerical results computed at each time step for printout.

A further description of the content and definition of each COMMON block can be found in Table A.3.2.2.

A.3.3 Core Requirements

The CONCRE program consists of 21 user supplied procedures using approximately 1800 cards (or lines). It also uses several FORTRAN and System Library routines. The program compiles in less than 65K words of memory and can be run either interactively or on batch.

A breakdown of the core required for each procedure is given in Table A.3.3.1. This shows a total of 2080 core positions. However, using labeled COMMON blocks causes the CDC 6600 NOS/BE1 Operating System to set aside a minimum of 52K memory positions for COMMON; so, the actual core required to run the program is 54K.

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS

Common Block Label	Variable	Length	Descriptive Title	Defined In
BARS (all real)	BARA	1	$\overline{A}$	CALBAR
	BARB	1	$\overline{B}$	CALBAR
	BARC	1	$\overline{C}$	CALBAR
	BARD	1	$\overline{D}$	CALBAR
	BARDEL	1	$\overline{JK}$ or $\overline{TU}$	CALBAR
	BARDNM	1	Denominator of $\overline{A},\overline{D},\overline{H},\overline{R}$	CALBAR
	BARG	1	$\overline{G}$	CALBAR
	BARH	1	$\overline{H}$	CALBAR
	BARJ	1	$\overline{J}$	CALBAR
	BARK	1	$\overline{K}$	CALBAR
	BARR	1	$\overline{R}$	CALBAR
	BARS	1	$\overline{S}$	CALBAR
	BART	1	$\overline{T}$	CALBAR
	BARTHE	1	$\overline{AB}$ or $\overline{RS}$	CALBAR
	BARU	1	$\overline{U}$	CALBAR
	EXPBX1	1	$\exp\beta(X-1)$	CALBAR
	EXPBZ1	1	$\exp\beta(zX-1)$	CALBAR
COMPS (all real)	ATHEDB	1	Constant part of θ	BTZERO
	ATHEDL	1	Constant part of $\ddot{\theta}$	BTZERO
	ATHED1	1	Constant part of $\ddot{\theta}$	PTZERO

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Continued

	ATHED2	1	Constant part of $\ddot{\theta}$	BTZERO, PTZERO
	AVDOT	1	Constant part of $\dot{V}(\dot{PE})$	BTZERO, PTZERO
	AWFDB	1	Constant part of $\dot{WF}$	BTZERO, PTZERO
	AWFDL	1	Constant part of $\dot{WF}$	BTZERO, PTZERO
	BTHED2	1	Constant part of $\ddot{\theta}$	BTZERO
	BTHED3	1	Constant part of $\ddot{\theta}$	BTZERO
	BVDOT	1	Constant part of $\dot{V}(\dot{PE})$	BTZERO
	BWFDOT	1	Constant part of $\dot{WF}$	BTZERO
	BWPDOT	1	Constant part of $\dot{WP}$	BTZERO
	PWFDB	1	Constant part of $\dot{WF}$	PTZERO
	PWFDL	1	Constant part of $\dot{WF}$	PTZERO
	PWPDOT	1	Constant part of $\dot{WP}$	PTZERO
	PVDOT	1	Constant part of $\dot{V}(\dot{PE})$	PTZERO
CONSTS (all real)	ACUBE	1	a^3	TZERO
	ARSQ	1	$\overline{AR}^2$	TZERO

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Continued

ARSZP1	1	$(\overline{AR}^2)(z)+1$	PTZERO
ARZSP1	1	$(\overline{AR})(z^2)+1$	PTZERO
ASQ	1	a^2	TZERO
B	1	b	BTZERO, PTZERO
BARA1	1	Constant part of $\overline{A}$	PTZERO
BARC1	1	Constant part of $\overline{C}$	PTZERO
BARG1	1	Constant part of $\overline{G}$	PTZERO
BARH1	1	Constant part of $\overline{H}$	PTZERO
BARJ1	1	Constant part of $\overline{J}$	PTZERO
BETACB	1	β^3	TZERO
BETAFR	1	β^4	TZERO
BETASQ	1	β^2	TZERO
DELTAK	1	nw/m	TZERO
EPSLNU	1	0.2	CONCRE
EXPBET	1	$\exp(-\beta)$	TZERO
FOURTH	1	1/4	CONCRE
HALF	1	1/2	CONCRE
ONEMZ	1	1-z	PTZERO
ONEPZ	1	1+z	PTZERO

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Continued

	SIXTH	1	1/6	CONCRE
	THETU1	1	$4h\varepsilon_u/d^2$	TZERO
	THIRD	1	1/3	CONCRE
	W	1	w=386.4m	TZERO
	XMU	1	M_u	TZERO
	Z	1	z	PTZERO
	ZCUBE	1	z^3	PTZERO
	ZFOUR	1	z^4	PTZERO
	ZSQ	1	z^2	PTZERO
FLAGS (all integer)	BADXFG	1	Bad-Hinge-Flag	TCNTRL, CHEKXH
	BPFLAG	1	Beam-or-Plate-Flag	NXTCAS, GETITL
	DONEFG	1	Case-is-Done-Flag	DRIVER, TCNTRL, TSTEP, CHEKXH
	EOFLAG	1	End-of-File-Flag	CONCRE, GETITL, RWDATA
	IERRFG	1	Input-Error-Flag	DRIVER, RWDATA
	KOUNT	1	System Input Error Count	CONCRE
	LOADFG	1	Type-of-Load-Flag	TCNTRL
	MECHFG	1	Mechanism-Flag	TCNTRL, CHEKXH

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Continued

	M1STFG	1	First-Time-thru-COMP-Flag	TCNTRL, COMP
	NCONFG	1	Non-Convergent-Flag	TCNTRL, PTZERO, CALXHO
INPUTS (all real)	A	1	a	RWDATA
	AIDA	1	n	RWDATA
	ALPHA	1	α	RWDATA
	AR	1	$\overline{AR}$	RWDATA
	BETA	1	β	RWDATA
	D	1	d	RWDATA
	F	1	F	RWDATA
	H	1	h	RWDATA
	PC	1	P_C	RWDATA
	PE	1	P_E	RWDATA
	Q	1	q	RWDATA
	SIGMAC	1	σ_c	RWDATA
	SIGMAR	1	σ_r	RWDATA
	TAU	1	τ	RWDATA
	TINCR	1	dt	RWDATA
	TMAX	1	t_{max}	RWDATA
	TPRINT	1	t_{print}	RWDATA
	XM	1	m	RWDATA
	WAVEFN	1	Pressure Wave Function Code	RWDATA

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Continued

PRINTS	FLAG	1	Failure-Flag (alphabetic, A1)	TCNTRL, TSTEP
	MAXLIN	1	Maximum lines/ page (integer)	CONCRE
	NAIDA	1	n+2 (integer)	RWDATA
	NF	1	F (integer)	RWDATA
	NUMLIN	1	Current line number on listing (integer)	PRINTR, PAGE
	NUMPAG	1	Current page number on listing (integer)	RWDATA, PAGE
	NWAVEF	1	Pressure Wave Function (integer)	RWDATA
	STEPCT	1	Count of integration steps since last printed line (integer)	TSTEP, PRINTR
	TIMNOW	1	Time of run (alphabetic, A9)	CONCRE
	TITLE	15	Case title (alphabetic, 15A5)	GETITL
	TODAY	1	Date of run (alphabetic, A10)	CONCRE
	TYPE	1	Type of Case (alphabetic, A5)	GETITL

TABLE A.3.2.2. COMMON BLOCK DESCRIPTIONS, Concluded

RESULT (all real)	DELTA	1	δ	TCNTRL, TSTEP, RUNGEK
	T	1	Current time step, t	TCNTRL, TSTEP
	THETA	1	θ	TCNTRL, RUNGEK
	THETAD	1	$\dot{\theta}$	TCNTRL, RUNGEK
	V	1	Potential Engergy, V(PE)	TCNTRL, RUNGEK
	VEL	1	Velocity, $\dot{\delta}$	TCNTRL, TSTEP, RUNGEK
	WF	1	WF	TCNTRL, RUNGEK
	WK	1	Kinetic Energy, WK(KE)	TCNTRL, TSTEP
	WP	1	WP	TCNTRL, RUNGEK
	X	1	X	TZERO, CHEKXH
	XH	1	Current hinge location, x_h	TZERO, CHEKXH

TABLE A.3.3.1. CONCRE CORE REQUIREMENTS BY PROCEDURE

Procedure Name	Program Length	Buffer Length	COMMON Length
CONCRE	192	948	66
DRIVER	12		10
NXTCAS	2		36
GETITL	29		36
RWDATA	241		55
TCNTRL	8		66
TZERO	112		70
BTZERO	5		75
PTZERO	23		92
CALXHO	52		29
BISECT	20		
FTNZ	19		59
BFTNX	13		59
PFTNX	7		66
TSTEP	52		96
RUNGEK	47		40
COMP	19		103
CALBAR	16		76
CHEKXH	43		40
PRINTR	19		37
PAGE	109		36
TOTALS	1040	948	129

TABLE A.3.3.2. CONCRE CORE REQUIREMENTS

Types of Cases	Procedures Eliminated	Core Required
Beam only	PTZERO, FTNZ, PFTNX, CALBAR	2052
Plate only	BTZERO, BFTNX	2099
Beam and Plate	None	2117

This program is set up to run both plate and beam cases. If one type is eliminated, some of the FORTRAN procedures may also be eliminated. Table A.3.3.2 shows this information along with the savings in core requirements.

A.3.4 Library Routines

As mentioned earlier, the program uses several System and FORTRAN Library Routines. Tables A.3.4.1 list these routines with a short description. It includes the names, in parenthesis, of the variables affected by these routines, as well as the CONCRE subprocedure(s) from which each is called.

A.4 Input and Output

Input to the program is in 80-column card format, using input unit number 5, while the only output of the program is a 133-character listing or print file, using output unit number 6.

A.4.1 Program Input Data

The six ordered input card formats shown in Table A.4.1.1 describe the parameters for a single case. This table includes the name, in parenthesis, of the program variable in which each piece of input data is stored.

TABLE A.3.4.1. DESCRIPTION OF SYSTEM AND FORTRAN LIBRARY ROUTINES USED

ABS(u)	provides the absolute value of u; a FORTRAN intrinsic function (EPSILN in BISECT)
DATE(u)	returns the current date as the value of u in alphabetic form (10Hbmm/dd/yyb); a FORTRAN utility subroutine (TODAY in CONCRE)
EOF(u)	tests for an end-of-file condition on file unit u, returning the real value 0.0 if not set or 1.0 if an end-of-file condition was found; a FORTRAN utility function (file 5 in GETITL and RWDATA)
ERRSET (u_1,u_2)	sets the maximum number of input errors to be allowed before program termination to u_2 and keeps the count of such errors in u_1; a FORTRAN utility subroutine (KOUNT in CONCRE)
EXP(u)	computes e^u; a FORTRAN external function (EXPBET in TZERO, EXPBZ1 in FTNZ, BFTNZ in BFTNZ, EXPBX1 and EXPBZX in CALBAR, and FTNT and EXPBCX in COMP)
PDEN(i)	returns the current print density setting (lines/inch) for the printer through the value of the argument i; a System Library routine (J, MAXLIN in CONCRE)

TABLE A.3.4.1. DESCRIPTION OF SYSTEM AND FORTRAN LIBRARY ROUTINES USED, Concluded

SIGN (u_1,u_2)	multiplies the sign of u_2 by the value of u_1; a FORTRAN intrinsic function (SF1, SF2, SF3 in BISECT)
SQRT (u)	takes the square root of u; a FORTRAN external function (THETAU in TSTEP)
TIME (u)	returns the current clock time as the value of the argument u in alphabetic form (9Hbhh.mm.ss); a FORTRAN utility subroutine (TIMNOW in CONCRE)

TABLE A.4.1.1. DESCRIPTION OF INPUT FOR CONCRE

Card 1: (16A5) TYPE, TITLE

Type of case, beginning in column 1, "BEAM " or "PLATE" (TYPE)

Identifying title for case, free-field, beginning in or after column 6, up to 75 characters allowed (TITLE)

Card 2: (4F12.1) A, H, AR, XM

Plate Half Width or Beam Half Span, a, in. (A)

Thickness of Plate or Beam, h, in. (H)

Length to Width Ratio, $\overline{AR}$, dimensionless (AR)

Mass per Unit Area, m, lbs. $sec^2/-in.^3$ (XM)

Card 3: (4F12.1) PC, PE, ALPHA, TAU

Maximum Distributed Pressure Load, P_C, psi (PC)

Uniform Pressure Load, P_E, psi (PE)

Pressure Decay Constant, α, dimensionless (ALPHA)

Duration of Pressure, τ, sec. (TAU)

Card 4: (4F12.1) F, WAVEFN, AIDA, BETA

Type of Support for Beam or Plate, F, dimensionless: (F)

1, Simply-Supported

2, Clamped-Supported

TABLE A.4.1.1. DESCRIPTION OF INPUT FOR CONCRE, Concluded

Type of Pressure Wave Function; dimensionless: (WAVEFN)

1, General Wave

2, Square Wave Function

Weight Vector, n, dimensionless: (AIDA)

0, Vertical Wall

1, Horizontal Slab, Explosive Below

-1, Horizontal Slab, Explosive Above

Spatial Pressure Decay Constant, β, dimensionless (BETA)

Card 5: (4F12.1) SIGMAC, SIGMAR, Q, D

Concrete Compressive Strength, σ_c, psi (SIGMAC)

Reinforced Steel Yield Stress, σ_r, psi (SIGMAR)

Reinforcement Ratio in Tension, q, dimensionless (Q)

Distance Between Tensile Reinforcing Rod and opposite edge, d, in. (D)

Card 6: (3F12.1) TINCR, TMAX, TPRINT

Integration Time Increment, dt, sec. (TINCR)

Integration Stop Time, t_{max}, sec. (TMAX)

Integration Time Interval between printed lines, t_{print}, sec. (TPRINT)

Any number of cases can be run one after the other by having the sets of six cards for each case immediately follow one another. Beam and Plate cases can be placed in the same input deck. Errors found in one case should not affect the running of any cases which follow it. (Exception: if a case has less than six input cards, the program will attempt to read some of the cards in the case which follows, and will skip over that case.

A.4.2 Program Output

Output from the program CONCRE is directed to the printer. It includes listing all input data as read, the results desired, and several informative/error messages. The output of a sample case is shown in Section 3.3.

Table A.4.2.1 shows the information included in a normal output listing, with the corresponding program variable names given in parenthesis. The information from the first input card (Type and Title) is included in all the page headings for the case. The numerical input data is printed out on the first output page, as well as some constants computed for the case. The second and succeeding pages show the computed results, at the time steps specified.

If errors are found in reading or computing initial case variables, only the first page is printed with the

TABLE A.4.2.1. CONCRE OUTPUT

Page Headings (printed from PAGE)

Type of Case, "BEAM " or "PLATE"	(TYPE)
Current Date, DD/MM/YY	(TODAY)
Time at start of run, HH.MM.SS	(TIMNOW)
Case Title	(TITLE)
Page Number	(NUMPAG)
Type of Support for Slab	(NF)
Type of Pressure Wave	(NWAVEF)
Position of Slab relative to Explosive	(NAIDA)
Type of Load	(LOADFG)

First Page (Input data printed from RWDATA)

a, in.	(A)
h, in.	(H)
Length to Width Ratio, dimensionless	(AR)
m, lbf.-sec.^{2}/in.^{3}	(XM)
P_C, psi	(PC)
P_E, psi	(PE)
α, dimensionless	(ALPHA)
τ, sec	(TAU)
Support Factor, F	(F)
Type of Pressure Wave	(WAVEFN)
Weight Vector, n	(AIDA)

TABLE A.4.2.1. CONCRE OUTPUT, Concluded

β, dimensionless	(BETA)
σ_c, psi	(SIGMAC)
σ_r, psi	(SIGMAR)
q, dimensionless	(Q)
d, in.	(D)
dt, sec.	(TINCR)
t_{max}, sec.	(TMAX)
t_{print}, sec.	(TPRINT)

Second and Following Pages (Results printed from PRINTR)

Current time step, t, sec.	(T)
Rotation, θ, rad.	(THETA)
Failure Flag ("*") indicating fracture of reinforcing element ($\theta > \theta_u$)	(FLAG)
Midpoint Velocity, $\dot{\delta}$, in./sec.	(VEL)
Midpoint Deflection, δ, in.	(DELTA)
Work done by Pressure Load, WF, in.-lbs.	(WF)
Plastic Energy, WP, in.-lbs.	(WP)
Kinetic Energy, WK, in.-lbs.	(WK)
Current Hinge Location, x_h, in.	(XH)

appropriate error messages. (It is possible to have so many error messages that they run into the second page, but no headings will be printed.) Other problems can be found once computations are underway; these will be printed as they occur, usually causing the case calculations to be terminated. A description of all the error messages is given in Table A.4.2.2. This includes the message itself, a description of the cause of the message with possible steps for the user to take, and the name of the FORTRAN subprocedure in parenthesis from which the message was printed.

A.5 Numerical Techniques

This section describes the numerical techniques used by the program, CONCRE. These are the bisection (or binary search) method used by the subroutine BISECT to find the roots of the equations defining X_0 and z and the fourth-order Runge-Kutta technique used to solve the five simultaneous differential equations of work and motion in the RUNGE-KUTTA function.

A.5.1 The Bisection Method

The bisection (or binary search) method uses the fact that a function which crosses the x-axis (or goes to zero) within a given interval must have different signs at the end points of that interval. Because of this assumption, this

TABLE A.4.2.2. CONCRE ERROR MESSAGES

1. INPUT ERRORS FOUND--CASE TERMINATED.
   Invalid or non-numeric data has been encountered while reading numeric data for this case. This message will follow other messages which should better specify the error(s). The program will proceed to the next case. (DRIVER)

2. BAD TYPE & TITLE CARD FOUND
   XXXXXXXXXXXXXXXXX . . . XXXXXX
   CARDS MAY BE OUT OF ORDER
   The program was trying to read the Type & Title (or first) card of a new case, but found a card which did not contain "BEAM " or "PLATE" in the first five columns. The card found is printed in the message and is considered erroneous. The program will try to read the next card as a Type & Title card. (GETITL)

3. END OF FILE FOUND DURING INPUT OF DATA
   CASE IS TERMINATED AS NOT ALL DATA IS PRESENT
   An end-of-file condition was encountered while the program was trying to read the numeric data (cards 2 through 6) for the given case. The case is terminated and the program will print Message 5 and begin normal termination. (RWDATA)

TABLE A.4.2.2. CONCRE ERROR MESSAGES, Continued

4. DATA ERROR(S) WERE FOUND IN INPUT

   CASE IS TERMINATED AS DATA IS IN INCORRECT FORM

   Non-numeric data has been encountered while reading input cards 2 through 6 for the given case. This message will follow System Error messages which should show the erroneous card(s). The program will print Message 5 and Message 1 and proceed to the next case. (RWDATA)

5. SOME OF THE VALUES LISTED ABOVE MAY ACTUALLY BE FROM PREVIOUS CASE

   The input data for the case has been printed on this same page. However, if invalid data was read, the System did not allow the new invalid value for the given variable to replace the value used by the last case. This message follows Message 3 or Message 4. (RWDATA)

6. VALUES OF AIDA, F, OR WAVEFN ARE OUT OF RANGE

   CASE IS TERMINATED.

   The correct values are 1, 0, or -1 for AIDA (the Weight Vector), 1 or 2 for F (the Support Factor), and 1 or 2 for WAVEFN (the Wave Function). One or more of these is not correct. The program will print Message 1, and begin on a new case. (RWDATA)

7. SOME COMPUTED CONSTANTS FOR THIS CASE ARE NEGATIVE

   SOME INPUT VALUES MUST BE IN ERROR

TABLE A.4.2.2. CONCRE ERROR MESSAGES, Continued

CASE IS TERMINATED

The values of B (Beam Width or Plate Half Length, b), XMU (Hinge Moment, M_u), and W (Weight per unit Area, w) are computed from input data values and should not be negative. Recheck input values. The program will proceed to the next case. (TZERO)

8. THE ROOT OF THE EQUATION FOR Z WAS NOT BETWEEN ZERO AND ONE

The root-finding procedure, BISECT, failed to find a root for z (the Ratio of the Final Hinge Location to the Plate Half Length) in the interval (0, 1). Some of the input data must be incorrect. The program proceeds to the next case. (PTZERO)

9. NO VALUE OF THE ORIGINAL HINGE LOCATION, XHO, WAS FOUND IN THE INTERVAL (0, A).

IT IS ASSUMED TO BE THE VALUE OF A.

The root-finding procedure, BISECT, was unable to find a value of x_{h0} in the interval (0, A). Our experience shows it is acceptable to assume the value of x_{h0} as a. (CALXHO)

10. THE BISECTION METHOD USED TO FIND THE ORIGINAL HINGE LOCATION, XHO, DID NOT CONVERGE AFTER 100 ITERATIONS.

THE RESULT OF THE LAST ITERATION WILL BE ASSUMED CORRECT: XHO = XXX.

TABLE A.4.2.2. CONCRE ERROR MESSAGES, Continued

The root-finding procedure, BISECT, failed to converge within the given ε (~0.00000001) after 100 iterations. While this could be caused by a discontinuity of the given function, no known discontinuities exist in the given interval for either the Beam or Plate x_{h_0} functions. The program assumes convergence is close, sets x_0 to the average of the last two values used by the Bisection method, and continues. (CALXH0)

11. TIME EXCEEDED

    The maximum integration time (TMAX) has been reached without finding the Maximum Deflection for the case. The program goes to the next case. Increase the integration stop time (TMAX) and rerun case to obtain further results. (TSTEP)

12. INSUFFICIENT PRESSURE TO GIVE A RESPONSE

    The Maximum Deflection was reached before the first integration time step was completed. The program goes to the next case. If input data looks correct, decrease value of the integration time step (TINCR) and rerun the case. (TSTEP)

13. HINGE LOCATION HAS OVERSHOT FINAL HINGE LOC

    FINAL HINGE LOC = XXX   HINGE IS AT XXX

TABLE A.4.2.2. CONCRE ERROR MESSAGES, Concluded

TIME INCREMENT HAS BEEN HALVED--CASE WILL BE RERUN

Between the last two integration time steps, the hinge location, x_h, moved from a Mechanism 2 position past the final hinge location (more than 2% past). This infers that the computations are too inexact. Hence, the program automatically divides the integration time step (TINCR) by two and reruns the case, starting again at t=0. This will only be done a maximum of once (MAXTRI = 2); then the case will be terminated. (CHEKXH)

14. HINGE LOCATION IS NEGATIVE--CASE IS TERMINATED

CHECK INPUT VALUES

The results of the last integration step have produced a negative hinge location ($x_h = \delta/\theta$). This makes no sense; data values must be wrong. (CHEKXH)

15. AN ASTERISK INDICATES THAT A REINFORCING ELEMENT HAS FRACTURED

This is a standard footnote printed on the second and succeeding pages of the output for every case. Should there be an asterisk following the value printed for θ, then θ is greater than θ_u which infers the failure of a reinforcing rod. (TSTEP, PAGE)

method will not work for finding roots of functions which only become tangent to the x-axis. Nor will it work for a function with an even number of roots within the given interval, since the signs of the function at the ends of the interval would be the same.

Basically, the method works in the following manner. First, it checks the signs of the function at the end points. If they are the same, the routine terminates in error. If they are not the same, the midpoint of the interval is chosen and the sign and value of the function at that point are determined. If the value is zero, the root has been found. If not, the interval is halved using the midpoint to replace the previous end point whose function had the same sign. The midpoint of the new interval is chosen and the method is repeated until the function or the interval goes to zero. In other words, the interval containing the root is made smaller and smaller until the root is the midpoint of that interval.

The three functions which require roots found by this method within the program are the beam equation for X_0 in the interval (0, 1) (the function BFTNX), the plate equation for X_0 in the interval (0,1) (the function PFTNX), and the equation for z used by plate cases in the interval [0, 1] (the function FTNZ). Tests run during the formation of the

program showed that each of these functions had one real root, at most, in the intervals given.

Many other root-finding techniques could have been used instead but the bisection method was chosen for its simplicity and rapid convergence.

A.5.2 The Fourth-Order Runge-Kutta Technique

The fourth-order Runge-Kutta technique is used to solve simultaneously at each time step the two second-order differential equations of motion ($\ddot{\theta}$ and $\ddot{\delta}$) with the three first order differential equations describing the work functions ($\dot{WF}$, $\dot{WP}$, and $\dot{V}(PE)$). These five equations are functions of time, t, the hinge location, x_h, and the velocity of rotation, $\dot{\theta}$ as follows:

$$\ddot{\delta} = \begin{cases} f_1(t,x_h), \text{ Mechanism 2} \\ 0, \text{ Mechanism 1} \end{cases}$$

$$\ddot{\theta} = f_2(t,x_h)$$

$$\dot{WF} = f_3(t,x_h,\dot{\theta})$$

$$\dot{WP} = f_4(\dot{\theta})$$

$$\dot{V}(PE) = f_5(x_h,\dot{\theta})$$

Now the hinge location, x_h, is defined in terms of θ and δ as:

$$x_h = \delta/\theta$$

So, our equations become, in Mechanism 2 (the more complex form):

$$\ddot{\delta} = f_1(t,\delta,\theta)$$
$$\ddot{\theta} = f_2(t,\delta,\theta)$$
$$\dot{WF} = f_3(t,\delta,\theta,\dot{\theta})$$
$$\dot{WP} = f_4(\dot{\theta})$$
$$\dot{V}(\dot{PE}) = f_5(\delta,\theta,\dot{\theta})$$

For the sake of uniformity, the subprocedures RUNGEK and COMP expect each of these five functions to have five arguments; i.e., $f_n(t,\delta,\dot{\delta},\theta,\dot{\theta})$, even though $\dot{\delta}$ is never used. When the Runge-Kutta computations for the time step have been completed, the new value of x_h is computed from the just-obtained values of θ and δ (in CHEKXH).

Hence, given the initial conditions and letting h be the time increment, dt (or TINCR), we can determine, inductively, using the fourth-order Runge-Kutta technique for simultaneous second-order equations,

$$\delta_{t+1} = \delta_t + h\dot{\delta}_t + (h/6)(a_0+a_1+a_2)$$
$$\dot{\delta}_{t+1} = \dot{\delta}_t + (1/6)(a_0+2a_1+2a_2+a_3)$$
$$\theta_{t+1} = \theta_t + h\dot{\theta}_t + (h/6)(b_0+b_1+b_2)$$
$$\dot{\theta}_{t+1} = \dot{\theta}_t + (1/6)(b_0+2b_1+2b_2+b_3)$$

where

$$a_0 = hf_1(t,\delta,\dot{\delta},\theta,\dot{\theta})$$
$$a_1 = hf_1(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta},\dot{\delta}+\tfrac{1}{2}a_0,\theta+\tfrac{1}{2}h\dot{\theta},\dot{\theta}+\tfrac{1}{2}b_0)$$
$$a_2 = hf_1(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta}+\tfrac{1}{4}ha_0,\dot{\delta}+\tfrac{1}{2}a_1,\theta+\tfrac{1}{2}h\dot{\theta}+\tfrac{1}{4}hb_0,\dot{\theta}+\tfrac{1}{2}b_1)$$
$$a_3 = hf_1(t+h,\delta+h\dot{\delta}+\tfrac{1}{2}ha_1,\dot{\delta}+a_2,\theta+h\dot{\theta}+\tfrac{1}{2}hb_1,\dot{\theta}+b_2)$$

$$b_0 = hf_2(t,\delta,\dot{\delta},\theta,\dot{\theta})$$

$$b_1 = hf_2(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta},\dot{\delta}+\tfrac{1}{2}a_0,\theta+\tfrac{1}{2}h\dot{\theta},\dot{\theta}+\tfrac{1}{2}b_0)$$

$$b_2 = hf_2(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta}+\tfrac{1}{4}ha_0,\dot{\delta}+\tfrac{1}{2}a_1,\theta+\tfrac{1}{2}h\dot{\theta}+\tfrac{1}{4}hb_0,\dot{\theta}+\tfrac{1}{2}b_1)$$

$$b_3 = hf_2(t+h,\delta+h\dot{\delta}+\tfrac{1}{2}ha_1,\dot{\delta}+a_2,\theta+h\dot{\theta}+\tfrac{1}{2}hb_1,\dot{\theta}+b_2)$$

The first-order differential equations are computed at the same time; e.g.,

$$WF_{t+1} = WF_t+(1/6)(c_0+2c_1+2c_2+c_3)$$

where

$$c_0 = hf_3(t,\delta,\dot{\delta},\theta,\dot{\theta})$$

$$c_1 = hf_3(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta},\dot{\delta}+\tfrac{1}{2}a_0,\theta+\tfrac{1}{2}h\dot{\theta},\dot{\theta}+\tfrac{1}{2}b_0)$$

$$c_2 = hf_3(t+\tfrac{1}{2}h,\delta+\tfrac{1}{2}h\dot{\delta}+\tfrac{1}{4}ha_0,\dot{\delta}+\tfrac{1}{2}a_1,\theta+\tfrac{1}{2}h\dot{\theta}+\tfrac{1}{4}hb_0,\dot{\theta}+\tfrac{1}{2}b_1)$$

$$c_3 = hf_3(t+h,\delta+h\dot{\delta}+\tfrac{1}{2}ha_1,\dot{\delta}+a_2,\theta+h\dot{\theta}+\tfrac{1}{2}hb_1,\dot{\theta}+b_2)$$

and where a_0, a_1, a_2, b_0, b_1, and b_2 are defined as above. The equations for WP and V(PE) are similar.

From the above, it is clear that the arguments for the five different equations are the same at each step, i=0,1,2,3. These are computed in the subprocedure RUNGEK and passed in the array ARG to the subprocedure COMP which actually uses these arguments to compute a_i, b_i, etc for the given step, i. These results are stored in the arrays A0, A1, A2, and A3 through the dummy parameter array AA.

The initial conditions for each differential equation is determined earlier in the program; they are all initialized to zero in the procedure TCNTRL. Since the hinge location, x_h,

is the result of a division of θ which would be zero at time zero, we alter the arguments in the array ARG given above to use the original hinge location, x_{h_0}, found by procedures CALXHO and BISECT, for the first two parts of the first Runge-Kutta time step; $f_n(t, x_{h_0}, \dot{\delta}, 1.0, \dot{\theta})$. This can be done as the two arguments affected, δ and θ, are used solely to compute x_h. The arguments for θ for the third and final parts of the first time step use the results from the first two parts of the first time step (stored in A0 and A1) and will be non-zero, so that x_h can be determined properly.

An alternate approach would have been to define each equation in three unknowns: $f_n(t, x_h, \dot{\theta})$. In this notation the equation for δ_{t+1} would contain a_0 through a_3 defined as follows:

$$a_0 = hf_1(t, x_h = \delta/\theta, \dot{\theta})$$

$$a_1 = hf_1(t+\tfrac{1}{2}h, (\delta+\tfrac{1}{2}h\dot{\delta})/(\theta+\tfrac{1}{2}h\dot{\theta}), \dot{\theta}+\tfrac{1}{2}b_0)$$

$$a_2 = hf_1(t+\tfrac{1}{2}h, (\delta+\tfrac{1}{2}h\dot{\delta}+\tfrac{1}{4}ha_0)/(\theta+\tfrac{1}{2}h\dot{\theta}+\tfrac{1}{4}hb_0), \dot{\theta}+\tfrac{1}{2}b_1)$$

$$a_3 = hf_1, (t+h, (\delta+h\dot{\delta}+\tfrac{1}{2}ha_1)/(\theta+h\dot{\theta}+\tfrac{1}{2}hb_1), \dot{\theta}+b_2),$$

for the same initial conditions. Again, care would have to be taken with x_h, during the first two parts of the first time step.

A.6. PROGRAM LISTING

```
      PROGRAM CONCRE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C         WRITTEN BY C. C. SCHAUBLE  FOR  C. A. ROSS
C           UNIVERSITY OF FLORIDA  GRADUATE ENGINEERING CENTER
C             EGLIN AIR FORCE BASE, FLORIDA
C               MARCH, 1979
C
C         THIS PROGRAM, CONCRE, COMPUTES THE DEFLECTION OF THE CENTER
C         OF A REINFORCED CONCRETE BEAM OR PLATE CAUSED BY A LINEAR
C         OR BLAST LOAD.
C
C         THIS MAIN PROCEDURE CALLS THE SYSTEM SUBPROCEDURES, DATE,
C         TIME, PDEN, AND ERRSET TO INITIALIZE THE CURRENT DATE,
C         TIME AND PRINT DENSITY AND TO OVERRIDE THE AUTOMATIC
C         TERMINATION CAUSED BY INVALID INPUT DATA.  IT ALSO CALLS
C         THE SUBPROCEDURE, DRIVER, TO HANDLE THE READING AND
C         PROCESSING OF EACH CASE.
C
C         TAPE5 IS THE NORMAL INPUT FILE IN 80-COLUMN CARD FORMAT.
C         TAPE6 IS THE LISTING OR PRINT FILE.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C          INITIALIZATION
C
C          GET DATE AND TIME
C          GET MAXIMUM NUMBER OF LINES PER PAGE
C          SET CONTROL TO HANDLE INPUT ERRORS
C          CLEAR END-OF-FILE FLAG
C
      CALL DATE(TODAY)
      CALL TIME(TIMNOW)
      CALL PDEN(J)
      MAXLIN = 9 * J
      CALL ERRSET(KOUNT,100)
      EOFLAG = 0
C
C         SET UP CONSTANTS
C
      HALF   = 1.0/2.0
      THIRD  = 1.0/3.0
      FOURTH = 1.0/4.0
      SIXTH  = 1.0/6.0
      EPSLNU = 0.2
C
```

```
C       DO WHILE END-OF-FILE FLAG IS CLEAR
C
  100 IF (EOFLAG.NE.0)   GO TO 500
      CALL DRIVER
      GO TO 100
C
C       CLEAN UP LOOSE ENDS
C
  500 WRITE (6,9900)
 9900 FORMAT(1H1)
C
      STOP
      END
```

```
      SUBROUTINE DRIVER
C
C          THIS SUBROUTINE DRIVES ONE CASE FROM BEGINNING (READING
C          OF DATA) TO END (COMPLETION OF EXECUTION OR ERRONEOUS
C          TERMINATION).   IT CALLS SUBROUTINES NXTCAS, TO HANDLE THE
C          READING OF THE DATA FOR THE NEXT CASE, AND TCNTRL, TO
C          CONTROL THE CASE THROUGH THE TIME STEP PROCESSING, FROM
C          INITIALIZING FOR TIME ZERO THROUGH PROCESSING EACH TIME
C          STEP UNTIL TIME-MAX OR UNTIL NORMAL COMPLETION.
C          IT IS CALLED BY THE MAIN PROGRAM, CONCRE.
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
C
C          CLEAR INPUT-ERROR-FLAG AND READ NEXT CASE
C
      IERRFG = 0
      CALL NXTCAS
C
C
C          CHECK INPUT OR END-OF-FILE ERRORS
C
C          IF END OF FILE FOUND
C
C             THEN TERMINATE CASE
C
      IF (EOFLAG.NE.0)  GO TO 2000
C
C             ELSE
C              IF INPUT ERROR FOUND
C
      IF (IERRFG.EQ.0)  GO TO 500
C
C              THEN PRINT MESSAGE AND PROCEED TO TERMINATE CASE
        WRITE (6,9901)
        GO TO 1999
C
C              ELSE CONTINUE
C
C              CLEAR CASE-IS-DONE-FLAG AND NUMBER-OF-TRIES-COUNT
C
  500   DONEFG = 0
        NTRIES = 0
C
C              DO WHILE CASE-IS-DONE-FLAG IS CLEAR
C
 1000   IF (.NOT. (DONEFG.EQ.0) )  GO TO 1999
C
          CALL TCNTRL (NTRIES)
          GO TO 1000
C
C              END WHILE (DONEFG)
C
C            END IF (IERRFG)
```

```
C
 1999 GO TO 2000
C
C          END IF (EOFLAG)
C
 2000 RETURN
C
C          FORMAT STATEMENT
C
 9901 FORMAT (1H0,*INPUT ERRORS FOUND--CASE TERMINATED*)
C
      END
```

```
      SUBROUTINE NXTCAS
C
C          THIS SUBROUTINE OBTAINS THE INPUT FOR THE NEXT CASE AND
C          SETS UP THE OUTPUT HEADINGS AND LISTS THE INPUT.
C
C          THE BEAM-PLATE-FLAG (BPFLAG) IS SET TO SHOW WHETHER THE CASE
C          IS A BEAM OR A PLATE CASE.  THE END-OF-FILE FLAG (EOFLAG)
C          IS SET IF AN END-OF-FILE INDICATOR IS FOUND WHEN DATA IS
C          EXPECTED.
C
C          THIS ROUTINE CALLS TWO SUBROUTINES:
C            GETITL:  TO LOOK FOR THE FIRST CARD OF INPUT FOR THE CASE,
C                     A TYPE&TITLE CARD (ALPHABETIC INPUT)
C            RWDATA:  TO READ AND PRINT OUT THE NUMERIC INPUT VALUES,
C                     AS WELL AS ANY INPUT ERRORS.
C
C          THIS SUBROUTINE IS CALLED BY THE ROUTINE, DRIVER.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, MISTFG, NCONFG
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
C
C           CLEAR THE BEAM-OR-PLATE FLAG
C
      BPFLAG = 0
C
C          DO WHILE THE END-OF-FILE AND BEAM-OR-PLATE FLAGS ARE CLEAR
C               (I.E., UNTIL THE FIRST CARD OF A NEW CASE IS READ)
C
  100 IF (.NOT. (BPFLAG.EQ.0 .AND. EOFLAG.EQ.0) )
     1          GO TO 1000
C
        CALL GETITL
        GO TO 100
C
C          END WHILE (BPFLAG AND EOFLAG)
C
C          IF THE END-OF-FILE HAS NOT YET BEEN REACHED
C
 1000 IF (EOFLAG.NE.0)   GO TO 5000
C
C           THEN READ AND PRINT DATA FOR THE NEW CASE
C
        CALL RWDATA
C
C           ELSE CONTINUE
C          END IF (NOT EOF)
C
 5000 RETURN
C
      END
```

```
      SUBROUTINE GETITL
C
C          THIS SUBROUTINE SEARCHES THE INPUT FILE FOR THE NEXT
C          CARD (OR RECORD) WHICH CONTAINS THE CHARACTERS "BEAM "
C          OR "PLATE" IN THE FIRST FIVE POSITIONS OF THE CARD,
C          INDICATING A TYPE & TITLE INPUT CARD, THE FIRST INPUT
C          CARD FOR A CASE.  WHEN SUCH A CARD IS FOUND, THE INPUT
C          INFORMATION IS STORED IN THE COMMON BLOCK, PRINTS, TO BE
C          USED LATER IN PRINTING OUTPUT HEADINGS FOR THE CASE.
C          ANY CARDS FOUND WITHOUT "BEAM " OR "PLATE" IN THE FIRST
C          FIVE COLUMNS ARE CONSIDERED ERRONEOUS OR OUT-OF-ORDER AND
C          ARE PRINTED OUT AS ERRORS.
C
C          THE END-OF-FILE FLAG IS SET IF AN END-OF-FILE INDICATOR
C          IS DETECTED.  THE BEAM-OR-PLATE-FLAG IS SET ACCORDING TO THE
C          TYPE OF TYPE&TITLE CARD FOUND.
C
C          THIS SUBROUTINE IS CALLED BY THE PROCEDURE NXTCAS.  IT
C          USES THE SYSTEM SUBPROCEDURE, EOF, TO CHECK FOR AN
C          END-OF-FILE CONDITION.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, MISTFG, NCONFG
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
      DATA  BEAM  /5HBEAM /,
     1      PLATE /5HPLATE/
C
C
C          READ NEXT RECORD
C
      READ (5,9900)  TYPE, TITLE
C
C          IF END-OF-FILE HAS BEEN REACHED
C
      IF (EOF(5).NE.1.0)  GO TO 1000
C           THEN SET END-OF-FILE FLAG
        EOFLAG = 1
        GO TO 9000
C
C           ELSE CONTINUE BY CHECKING TYPE
C            CASE ON TYPE
C
C             IF THIS IS A BEAM CASE
 1000   IF (TYPE.NE.BEAM)  GO TO 2000
C
C                THEN SET BEAM-PLATE FLAG TO INDICATE BEAM (1)
             BPFLAG = 1
             GO TO 9000
C
C             IF THIS IS A PLATE CASE
 2000   IF (TYPE.NE.PLATE)  GO TO 3000
C
```

```
C             THEN SET BEAM-PLATE FLAG TO INDICATE PLATE (2)
          BPFLAG = 2
          GO TO 9000
C
C             ELSE THIS IS A BAD TYPE&TITLE CARD--WRITE MESSAGE
 3000   WRITE (6,9901)   TYPE, TITLE
C
C            END CASE ON TYPE
C
C           END IF (EOF)
C
 9000 RETURN
C
C           FORMAT STATEMENTS
C
 9900 FORMAT (16A5)
 9901 FORMAT (1H1/1H0/1H0,*BAD TYPE & TITLE CARD FOUND:*/5X, 16A5/
    1         * CARDS MAY BE OUT OF ORDER*)
C
      END
```

```
      SUBROUTINE RWDATA
C
C          THIS SUBROUTINE READS DATA CARDS # 2 THRU 6 FOR THE
C          CURRENT CASE AND PRINTS THE INPUT DATA OUT ON THE
C          LISTING.  IF INPUT ERRORS OCCUR, THEY ARE COUNTED AND
C          THE NUMBER OF ERRORS IS PRINTED AFTER THE LISTING OF
C          THE INPUT VALUES.  IF AN END-OF-FILE IS FOUND, AN ERROR
C          MESSAGE IS PRINTED AND THE EOF-OF-FILE FLAG IS SET.
C
C          THIS ROUTINE ALSO USES THE NEWLY-READ VALUES OF AIDA, F,
C          AND WAVE-FUNCTION-CODE, IF THEY ARE VALID, TO SET UP
C          SUBSCRIPTS FOR THE TYPE OF CASE DESCRIPTION TO BE INCLUDED
C          IN THE OUTPUT PAGE HEADINGS.
C
C          ALL INPUT DATA READ IS PLACED IN THE COMMON BLOCK, INPUTS.
C          THE PRINT HEADING SUBSCRIPTS ARE PLACED IN THE COMMON BLOCK,
C          PRINTS.
C
C          THIS SUBROUTINE IS CALLED BY THE PROCEDURE,NXTCAS.  IT
C          USES THE SYSTEM SUBPROCEDURE, EOF, TO CHECK FOR AN
C          END-OF-FILE CONDITION.  IT CALLS THE SUBPROCEDURE, PAGE, TO
C          SET UP THE HEADINGS ON A FRESH PAGE FOR THE NEW CASE.
C
C
      INTEGER   BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
C
C
C          SET UP OUTPUT HEADINGS ON A NEW PAGE AND
C          INITIALIZE COUNT OF INPUT ERRORS TO LAST COUNT
C
      NUMPAG = 0
      CALL PAGE
      LCOUNT = KOUNT
C
C          READ REMAINING CARDS FOR CASE
C           AND PRINT OUT INPUT VALUES
C
      READ (5,9910) A, H, AR, XM,
     1              PC, PE, ALPHA, TAU,
     2              F, WAVEFN, AIDA, BETA,
     3              SIGMAC, SIGMAR, Q, D,
     4              TINCR, TMAX, TPRINT
      WRITE (6,9911) A, H, AR, XM,
     1               PC, PE, ALPHA, TAU,
     2               F, WAVEFN, AIDA, BETA
      WRITE (6,9912)  SIGMAC, SIGMAR, Q, D,
     1                TINCR, TMAX, TPRINT
C
```

```
C          IF END-OF-FILE FOUND
C
      IF (.NOT. (EOF(5).NE.0.0) )  GO TO 1000
C
C            THEN SET END-OF-FILE FLAG  AND  PRINT ERROR MESSAGE
C
         EOFLAG = 1
         WRITE (6,9902)
         WRITE (6,9904)
C
C            ELSE CONTINUE
C          END IF (EOF)
C
C          IF ANY INPUT ERRORS WERE FOUND
C
 1000 NERR = KOUNT - LCOUNT
      IF (.NOT. (NERR.NE.0) )  GO TO 2000
C
C            THEN SET INPUT-ERROR FLAG  AND  PRINT ERROR MESSAGE
C
         IERRFG = 1
         WRITE (6,9903) NERR
         WRITE (6,9904)
C
C            ELSE CONTINUE
C          END IF (IERR)
C
C          IF AIDA, F, OR WAVEFN ARE WITHIN RANGE
C
 2000 IF (.NOT. (AIDA.GE.-1.0 .AND. AIDA.LE.1.0) .OR.
     1    .NOT. (F.EQ.1.0 .OR. F.EQ.2.0)           .OR.
     2    .NOT. (WAVEFN.EQ.1.0 .OR. WAVEFN.EQ.2.0) )  GO TO 2500
C
C            THEN SET UP PRINTING SUBSCRIPTS
C
         NAIDA = AIDA + 2.0
         NF = F
         NWAVEF = WAVEFN
         GO TO 4000
C
C            ELSE SET ERROR FLAG AND PRINT ERROR MESSAGE
C
 2500    IERRFG = 1
         WRITE (6,9905)
C
C          END IF (AIDA, F, WAVEFN)
C
 4000 RETURN
C
C           FORMAT STATEMENTS
C
 9902 FORMAT (1H0/1H0,*END OF FILE FOUND DURING INPUT OF DATA*/
     1        *CASE IS TERMINATED AS NOT ALL DATA IS PRESENT*)
 9903 FORMAT (1H0/1H0, I4, * DATA ERROR(S) WERE FOUND IN INPUT*/
     1        * CASE IS TERMINATED AS DATA IS IN INCORRECT FORM*)
```

```
9904 FORMAT (1H0, *SOME OF THE VALUES LISTED ABOVE MAY ACTUALLY BE*,
    1       * FROM PREVIOUS CASE*)
9905 FORMAT (1H0/1H0,*VALUES OF AIDA, F, OR WAVEFN ARE OUT OF RANGE*/
    1       * CASE IS TERMINATED*)
9910 FORMAT ( 4(4F12.0 /), 3F12.0)
9911 FORMAT (1H0,
    1  6X,51HPLATE HALF WIDTH OR BEAM HALF SPAN, IN.             (A),G15.8/
    2  7X,51HBEAM OR PLATE THICKNESS, IN.                        (H),G15.8/
    3  7X,51HLENGTH TO WIDTH RATIO, DIMENSIONLESS               (AR),G15.8/
    4  7X,51HMASS PER UNIT AREA, LBS-SEC.SQD/IN.CUBED           (XM),G15.8/
    5  1H0,
    6  6X,51HMAXIMUM DISTRIBUTED PRESSURE LOAD, PSI.            (PC),G15.8/
    7  7X,51HUNIFORM PRESSURE LOAD, PSI.                        (PE),G15.8/
    8  7X,51HPRESSURE DECAY, DIMENSIONLESS                   (ALPHA),G15.8/
    9  7X,51HPRESSURE DURATION, SEC.                           (TAU),G15.8/
    A  1H0,
    B  6X,51HSUPPORT FACTOR: 1=SIMPLY, 2=CLAMPED                 (F),G15.8/
    C  7X,51HWAVE FUNCTION: 1=GENERAL, 2=SQUARE             (WAVEFN),G15.8/
    D  7X,51HWEIGHT VECTOR: 0=VERT, 1=EXP BLW, -1=EXP ABV (AIDA),G15.8/
    E  7X,51HSPATIAL PRESSURE DECAY CONSTANT,DIMENSIONLESS(BETA),G15.8)
9912 FORMAT (1H0,
    1  6X,51HCONCRETE COMPRESSIVE STRENGTH, PSI.           (SIGMAC),G15.8/
    2  7X,51HREINFORCED STEEL YIELD STRESS, PSI.           (SIGMAR),G15.8/
    3  7X,51HREINFORCEMENT RATIO IN TENSION, DIMENSIONLESS    (Q),G15.8/
    4  7X,51HREINFORCING DISTANCE, IN.                          (D),G15.8/
    5  1H0,
    6  6X,51HTIME INCREMENT, SEC.                           (TINCR),G15.8/
    7  7X,51HTIME MAXIMUM, SEC.                              (TMAX),G15.8/
    8  7X,51HTIME STEP INTERVAL PER PRINTED LINE, SEC.  (TPRINT),G15.8)
C
      END
```

```
      SUBROUTINE TCNTRL (NTRIES)
C
C          THIS SUBROUTINE STARTS WITH EACH CASE AT TIME T=0 AND
C          AFTER SOME INITIALIZATION, PUSHES THE CASE THROUGH THE
C          TIME STEPS, USING TINCR AS THE TIME STEP INCREMENT AND
C          TMAX, AS THE TIME STOP INDICATOR.
C
C          THE BAD-HINGE-LOCATION-FLAG IS SET IF A BAD VALUE FOR THE
C          HINGE LOCATION IS DISCOVERED BY THE SUBPROCEDURE, CHEKXH.
C          THE LOOP-CONTROL-FLAG IS SET WHEN THE SUBPROCEDURE, TSTEP,
C          FINDS THE TIME-STEP-LOOP SHOULD BE TERMINATED, FOR ANY
C          REASON.  THE CASE-IS-DONE-FLAG CAN BE SET IN THE FOLLOWING
C          WAYS:
C            1:  IF THE TIME-STEP-LOOP HAS BEEN ATTEMPTED MAXTRI
C                TIMES (SET IN TCNTRL)
C            2:  IF THE ORIGINAL CONSTANTS FOR THE CASE ARE NEGATIVE
C                OR WOULD NOT CONVERGE (SET IN TZERO)
C            3:  IF THE MAXIMUM DEFLECTION HAS BEEN FOUND OR IF THE
C                TIME MAXIMUM HAS BEEN EXCEEDED (SET IN TSTEP).
C
C          THE PARAMETER, NTRIES, IS INCREMENTED BY THIS ROUTINE,
C          WHENEVER THE TIME-STEP-LOOP IS TERMINATED, TO INDICATE THE
C          NUMBER OF TIMES THE LOOP HAS BEEN TRIED.  IN THE CASE OF
C          AN UNSUCCESSFUL TERMINATION, THE LOOP WILL BE TRIED AGAIN,
C          UP TO MAXTRI TIMES, WITH THE TIME STEP HALVED EACH TIME.
C
C          THIS SUBROUTINE CALLS THE FOLLOWING SUBPROCEDURES:
C            TZERO:    TO INITIALIZE CONSTANTS AND VARIABLES FOR THE CASE
C            TSTEP:    TO COMPUTE THE DESIRED VARIABLES AT EACH TIME STEP,
C                      FROM TIME = TINCR THROUGH THE TIME, T = TMAX.
C
C          THIS ROUTINE IS CALLED BY THE PROCEDURE, DRIVER.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
      DATA   MAXTRI /2/
      DATA   BLANK /1H /
C
C
C          SET MECHANISM-FLAG TO USE MECHANISM 2 TO CHECK XHO,
C           CLEAR BAD-HINGE-LOCATION AND LOOP-CONTROL FLAGS,
C           SET THE FIRST-TIME-FLAG FOR THE FIRST TWO PASSES THRU COMP,
C           CLEAR NON-CONVERGENT-FLAG FOR PLATE CASE,
C           CLEAR FAILURE FLAG FOR PRINTOUT,
C           AND SET TYPE-OF-LOAD FLAG TO BLAST LOAD UNLESS BETA=0.
C           ALSO, INITIALIZE TIME VARIABLES, COMPUTE CASE CONSTANTS
```

```
C           FROM INPUT AND PRINT T=0 RESULTS ON NEW PAGE (TZERO).
C
      MECHFG = 2
      BADXFG = 0
      LOOPFG = 0
      M1STFG = 2
      NCONFG = 0
      LOADFG = 2
      IF (BETA.EQ.0.0)  LOADFG = 1
C
      T       = 0.0
      THETA   = 0.0
      THETAD  = 0.0
      DELTA   = 0.0
      VEL     = 0.0
      WF      = 0.0
      WP      = 0.0
      V       = 0.0
      WK      = 0.0
C
      FLAG = BLANK
      NUMLIN = MAXLIN
      CALL TZERO (NTRIES)
C
C          IF THE ORIGINAL CONSTANTS WERE CALCULATED SUCCESSFULLY
C
      IF (DONEFG.EQ.1)  GO TO 5000
C
C            THEN
C            DO WHILE LOOP-CONTROL-FLAG IS CLEAR
C              (I.E., WHILE TIME<TIME MAX  AND BADXFG IS CLEAR)
C
 1000   IF (LOOPFG.NE.0)  GO TO 4000
C
          CALL TSTEP (LOOPFG)
          GO TO 1000
C
C            END WHILE
C
C            ADD 1 TO NUMBER OF TRIES
C
 4000   NTRIES = NTRIES + 1
C
C            IF THIS WAS THE LAST TRY ALLOWED, SET CASE-IS-DONE-FLAG
C
        IF (NTRIES.GE.MAXTRI)  DONEFG = 1
C
C            ELSE CONTINUE, SKIPPING CALCULATIONS
C
C          END IF (ORIGINAL CONSTANTS)
C
 5000 RETURN
C
      END
```

```
      SUBROUTINE TZERO (NTRIES)
C
C          THIS SUBROUTINE COMPUTES AND PRINTS CASE CONSTANTS FROM THE
C          GIVEN INPUT VALUES.  IT ALSO PROVIDES FOR THE FIRST LINE OF
C          OUTPUT (T=0) TO BE PRINTED OUT ON A NEW PAGE WITH HEADINGS.
C
C          THIS SUBROUTINE IS CALLED BY THE PROCEDURE TCNTRL AND CALLS
C          THE FOLLOWING SUBPROCEDURES:
C            BTZERO:  TO INITIALIZE BEAM CONSTANTS AND VARIABLES
C            PTZERO:  TO INITIALIZE PLATE CONSTANTS AND VARIABLES
C            CALXH0:  TO FIND THE ORIGINAL HINGE LOCATION, XH0
C            CHEKXH:  TO CHECK THE ORIGINAL LOCATION OF THE HINGE,
C                     XH0, AT TIME = 0
C            PRINTR:  TO PRINT OUT THE INITIAL VALUES FOR THE GIVEN
C                     CASE AT TIME = 0
C            EXP:     THE SYSTEM LIBRARY FUNCTION FOR EXPONENTIALS
C
C          THE INPUT PARAMETER, NTRIES, IS USED TO PREVENT THE CALCU-
C          LATION AND PRINTING OF THE COMPUTED CONSTANTS FOR THE CASE
C          ON THE SECOND AND SUCCEEDING TRIES.  THE PARAMETER WILL
C          NOT BE ALTERED IN ANY WAY.
C
C          THE CASE-IS-DONE-FLAG MAY BE SET IN TWO WAYS:  FIRST,
C          WHEN THE PLATE CONSTANT, Z, CANNOT BE FOUND OR DOES NOT
C          CONVERGE, AND SECOND, IF NEGATIVE CONSTANTS ARE COMPUTED.
C
C
      INTEGER   BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C
C        IF THIS IS THE FIRST TIME THRU TZERO FOR THIS CASE
C
      IF (NTRIES.NE.0)  GO TO 3000
C
C              THEN USE INPUT VALUES TO COMPUTE CONSTANTS
C
        XMU = 0.9 * D**2 * Q * SIGMAR *
     1            (1.0 - 0.59 * Q * (SIGMAR/SIGMAC) )
        ASQ = A * A
        ACUBE = A**3
        ARSQ = AR*AR
        BETASQ = BETA*BETA
        BETACB = BETA**3
```

```
      BETAFR = BETA**4
      EXPBET = EXP(-BETA)
      W = 386.4 * XM
      DELTAK = AIDA * 386.4
      THETU1 = (4.0 * H * EPSLNU) / (D**2)
C
C          CASE ON TYPE OF CASE
C
C            IF BEAM CASE
C
      IF (BPFLAG.EQ.1)  CALL BTZERO
C
C            IF PLATE CASE
C
      IF (BPFLAG.EQ.2)  CALL PTZERO
C
C           END CASE ON TYPE
C
C          IF Z CALCULATIONS DID NOT CONVERGE
C
      IF (NCONFG.NE.1)  GO TO 1000
C
C            THEN SET CASE-IS-DONE-FLAG TO TERMINATE CASE
C
        DONEFG = 1
        GO TO 3000
C
C            ELSE FIND VALUE OF ORIGINAL HINGE LOCATION
C             AND PRINT OUT COMPUTED CONSTANTS
C
 1000   CALL CALXHO (XO, XHO)
C
        IF (BPFLAG.EQ.1)  WRITE (6,9901)  B, XMU, W, XHO
        IF (BPFLAG.EQ.2)  WRITE (6,9902)  B, Z, XMU, W, XHO
C
C            IF ANY NEGATIVE CONSTANTS WERE COMPUTED
C
        IF (.NOT. (B   .LE.0.0  .OR.
     1             XMU.LE.0.0  .OR.
     2             W   .LE.0.0      ) )  GO TO 3000
C
C              THEN PRINT ERROR MESSAGE AND TERMINATE CASE
C
          WRITE (6,9903)
          DONEFG = 1
C
C              ELSE CONTINUE
C            END IF (NEGATIVE CONSTANTS)
C
C          END IF (Z CONVERGED)
C
C         ELSE CONTINUE
C        END IF (1ST TIME)
C
C        IF CASE CONSTANTS APPEAR OKAY
```

```
C
 3000 IF (DONEFG.NE.0)  GO TO 4000
C
C             THEN INITIALIZE X AND XH TO ORIGINAL HINGE LOCATION,
C                CHECK MECHANISM AND PRINT TIME ZERO RESULTS ON NEW PAGE
C
         X  = X0
         XH = XH0
         CALL CHEKXH
         CALL PRINTR
C
C              ELSE CONTINUE
C           END IF (CONSTANT OKAY)
C
 4000 RETURN
C
C           FORMAT STATEMENTS
C
 9901 FORMAT (1H0/ 1H0, 2X, *COMPUTED CONSTANT VALUES* / 1H0,
     1   6X,51HBEAM WIDTH, IN.                                     (B),G15.8/
     2   7X,51HHINGE MOMENT, IN.-LBS./IN.                        (XMU),G15.8/
     3   7X,51HWEIGHT PER UNIT AREA, LBS./IN.SQ.                   (W),G15.8/
     4   7X,51HORIGINAL HINGE LOCATION, IN.                       (XH),G15.8)
 9902 FORMAT (1H0/ 1H0, 2X, *COMPUTED CONSTANT VALUES*/ 1H0,
     1   6X,51HPLATE HALF LENGTH, IN.                              (B),G15.8/
     2   7X,51HRATIO OF FINAL HINGE LOC TO B, DIMENSIONLESS        (Z),G15.8/
     3   7X,51HHINGE MOMENT, IN.-LBS./IN.                        (XMU),G15.8/
     4,  7X,51HWEIGHT PER UNIT AREA, LBS./IN.SQ.                   (W),G15.8/
     5   7X,51HORIGINAL HINGE LOCATION, IN.                       (XH),G15.8)
 9903 FORMAT (1H0/ 1H0, *SOME COMPUTED CONSTANTS FOR THIS CASE *
     1   *ARE NEGATIVE.*/ 1X, *SOME INPUT VALUES MUST BE IN ERROR.*/
     2   1X, *CASE IS TERMINATED.*/)
C
      END
```

```
      SUBROUTINE BTZERO
C
C         THIS SUBROUTINE CALCULATES THE INITIAL VALUES OF CONSTANTS
C         AND VARIABLES TO BE USED BY THE GIVEN BEAM CASE.
C
C         ALL OF THE CONSTANTS AND VARIABLES INITIALIZED ARE PASSED
C         TO THE OTHER PROCEDURES THROUGH THE COMMON BLOCKS, COMPS
C         AND CONSTS.
C
C         THIS SUBROUTINE IS CALLED BY THE PROCEDURE, TZERO.  IT CALLS
C         NO SUBPROCEDURES.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /COMPS /  ATHEDB, ATHEDL, ATHED1, ATHED2, AVDOT, AWFDB,
     1                 AWFDL, BTHED2, BTHED3, BVDOT, BWFDOT, BWPDOT,
     2                 PWFDB, PWFDL, PWPDOT, PVDOT
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C
C          SET UP BEAM CONSTANTS
C
      B = 2.0 * A / AR
      BTHED2 = 3.0 * HALF * DELTAK / A
      BTHED3 = 3.0 * F * XMU / (XM * ACUBE)
      BWFDOT = 2.0 * ASQ * B
      BWFDOT = 2.0 * B * F * XMU
      BVDOT = 2.0 * ASQ * B * AIDA * W
      ATHED2 = BTHED2 + BTHED3
      AVDOT = BVDOT * HALF
C
C         IF THIS IS A LINEAR LOAD
C
      IF (LOADFG.NE.1)  GO TO 2000
C
C            THEN DETERMINE LINEAR LOAD CONSTANTS FOR COMP
C
        ATHEDL = (3.0*PE + 2.0*PC) / (2.0 * A * XM)
        AWFDL = BWFDOT * (PE*HALF   +   PC*THIRD)
        GO TO 3000
C
C            ELSE DETERMINE BLAST LOAD CONSTANTS FOR COMP
C
 2000   ATHEDB = (3.0/(XM*A))*(PE*HALF
     1                         + (PC/BETACB)*(BETASQ - 2.0*BETA + 2.0
     2                                          - 2.0*EXPBET) )
```

```
      AWFDB = BWFDOT * (PE*HALF
     1                + (PC/BETACB)*(2.0 - BETA - 2.0*EXPBET
     2                                  + BETA*(BETA - 1.0) ) )
C
C         END IF (LINEAR LOAD)
C
 3000 RETURN
C
      END
```

```
      SUBROUTINE PTZERO
C
C          THIS SUBROUTINE COMPUTES INITIAL VALUES OF VARIABLES AND
C          CONSTANTS TO BE USED BY THE GIVEN PLATE CASE.  THE PLATE
C          CONSTANT Z IS FOUND USING THE ROOT-FINDING ROUTINE, BISECT,
C          WITH THE Z FUNCTION, FTNZ.  A MESSAGE IS PRINTED AND THE
C          CASE IS TERMINATED IF Z CANNOT BE FOUND.
C
C          THIS SUBROUTINE IS CALLED BY THE PROCEDURE, TZERO, AND
C          CALLS THE FOLLOWING SUBPROCEDURES:
C            BISECT:  TO SOLVE FOR Z (THE FUNCTION, FTNZ, IS CARRIED TO
C                     BISECT AS A PARAMETER)
C            CALBAR:  TO SET UP EQUATION OF MOTION CONSTANTS FOR
C                     MECHANISM 1.
C
C          THE NON-CONVERGENT-FLAG IS RETURNED TO THIS ROUTINE BY
C          THE ROUTINE, BISECT, AS SET, IF THE ROOT FOR THE Z FUNCTION
C          CANNOT BE FOUND.  THIS IS PASSED THROUGH THE COMMON BLOCK,
C          FLAGS, BACK TO THE CALLING PROCEDURE, TZERO.
C
C          ALL THE CONSTANTS AND VARIABLES INITIALIZED ARE STORED AND
C          PASSED TO THE OTHER PROCEDURES BY THE COMMON BLOCKS, BARS,
C          COMPS, AND CONSTS.
C
C
      EXTERNAL  FTNZ
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /BARS  /  BARA, BARB, BARC, BARD, BARDEL, BARDNM, BARG,
     1                 BARH, BARJ, BARK, BARR, BARS, BART, BARTHE, BARU,
     2                 EXPBX1, EXPBZX
      COMMON /COMPS /  ATHEDB, ATHEDL, ATHED1, ATHED2, AVDOT, AWFDB,
     1                 AWFDL, BTHED2, BTHED3, BVDOT, BWFDOT, BWPDOT,
     2                 PWFDB, PWFDL, PWFDOT, PVDOT
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C
C         SOLVE FOR Z
C
      NCONFG = 0
      CALL BISECT (-0.0001, 1.0001, Z, NCONFG, FTNZ)
C
C         IF Z IS NOT IN THE INTERVAL (0,1)
C
      IF (NCONFG.EQ.0)  GO TO 1200
C
C            THEN PRINT MESSAGE AND TERMINATE CASE
```

```
C
      WRITE (6,9901)
      GO TO 3000
C
C           ELSE CONTINUE ON
C
 1200 B      = A * AR
      ZSQ    = Z * Z
      ZCUBE  = Z**3
      ZFOUR  = Z**4
      ARSZP1 = ARSQ * Z   +   1.0
      ARZSP1 = AR * ZSQ   +   1.0
      ONEMZ  = 1.0 - Z
      ONEPZ  = 1.0 + Z
C
      BARA1 = 1.0 / (ZSQ * BETAFR * XM * A)
      BARC1 = - DELTAK / A
      BARG1 = - (F * XMU) / (XM * ACUBE * AR)
      BARH1 = 1.0 + AR
      BARJ1 = 1.0 / (XM * BETACB)
C
      CALL CALBAR(1.0)
      PWPDOT = 4.0 * F * XMU * A * ARSZP1 / (Z * AR)
      PVDOT  = 4.0 * AIDA * W * ASQ * B
      ATHED1 = BARTHE
      ATHED2 = BARC * BARD   +   BARG * BARH
      AVDOT  = PVDOT * (1.0 - ONEPZ*HALF + Z*THIRD)
C
C           IF THIS IS A LINEAR LOAD
C
      IF (LOADFG.NE.1)   GO TO 2000
C
C              THEN DETERMINE LINEAR LOAD CONSTANTS FOR COMP
C
         PWFDL = 4.0 * AR * ACUBE
         AWFDL = PWFDL
     1           * (PE * (1.0 - ONEPZ*HALF + Z*THIRD)
     2           + PC *(THIRD - ONEPZ*SIXTH + SIXTH - Z*ONEPZ*FOURTH) )
         GO TO 3000
C
C              ELSE DETERMINE BLAST LOAD CONSTANTS FOR COMP
C
 2000    PWFDB = 4.0 * ASQ * B / (ZSQ * BETAFR)
         AWFDB = PWFDB
     1           * (PE*BETAFR*ZSQ*(1.0 - Z*HALF + ONEPZ*THIRD)
     2              + PC*(2.0*BETASQ*ZSQ  -  4.0*BETA*ZSQ
     3                   + EXPBZX*(- 3.0*BETA*Z + 2.0*BETA*Z + 3.0)
     4                   + EXPBX1*ZSQ*(BETASQ - 5.0*BETA
     5                                 - BETASQ + 2.0*BETA + 9.0)
     6                   - EXPBET*(2.0*BETA*Z*ONEPZ + 9.0*ZSQ + 3.0) ) )
C
C              END IF (LINEAR LOAD)
C
C           END IF (BAD Z)
C
```

```
3000 RETURN
C
C          FORMAT STATEMENTS
C
 9901 FORMAT (/1H0, *THE ROOT OF THE EQUATION FOR Z WAS NOT*,
     1         *BETWEEN ZERO AND ONE*)
C
      END
```

```
      SUBROUTINE CALXHO (XO, XHO)
C
C          THIS SUBROUTINE HANDLES CALLING THE METHOD FOR SOLVING
C          WHATEVER FUNCTION IS NECESSARY TO FIND THE ORIGINAL HINGE
C          LOCATION FOR THE CASE--OR PRINTING AN ERROR MESSAGE IF
C          IT CANNOT BE FOUND EXACTLY.
C
C          THIS ROUTINE IS CALLED BY THE SUBROUTINE, TZERO.
C          THE ONLY SUBPROCEDURE CALLED BY THIS ROUTINE IS THE
C          ROOT-FINDING SUBPROCEDURE, BISECT, WHICH USES THE
C          FUNCTION, BFTNX OR PFTNX, DEPENDING OF THE TYPE OF CASE.
C
C          THE PARAMETERS RELATING TO THE COMPUTED HINGE LOCATION,
C          XO AND XHO, ARE RETURNED TO THE CALLING PROCEDURE, TZERO.
C
C
      EXTERNAL  BFTNX, PFTNX
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
C
C
C        COMPUTE XHO IN THE INTERVAL (0,A)
C
C          DETERMINE TYPE OF CASE FOR CORRECT XHO FUNCTION
C
C         IF BEAM CASE
      IF (BPFLAG.EQ.1)
     1  CALL BISECT (0.00001, 0.99999, XO, NCONFG, BFTNX)
C
C         IF PLATE CASE
      IF (BPFLAG.EQ.2)
     1  CALL BISECT (0.00001, 0.99999, XO, NCONFG, PFTNX)
C
C          END CASE OF TYPE OF CASE FOR XHO CALCULATION
C
      XHO = XO * A
C
C           CHECK CONVERGENCE
C
C            CASE 1: BISECT DID NOT FIND A ROOT IN THE INTERVAL (0,A)
C
      IF (NCONFG.NE.1)  GO TO 2100
C            ASSUME  XHO = A
        WRITE (6,9903)
        NCONFG = 0
        XO = 1.0
        XHO = A
C
C            CASE 2: BISECT DID NOT CONVERGE AFTER 100 ITERATIONS
C
 2100 IF (NCONFG.NE.-1)  GO TO 3000
```

```
C               ASSUME  ROOT FOUND IS CORRECT
      WRITE (6,9904)  XH0
      NCONFG = 0
C
C             END CASE ON CONVERGENCE
C
 3000 RETURN
C
C          FORMAT STATEMENTS
C
 9903 FORMAT (/1H0,*NO VALUE OF THE ORIGINAL HINGE LOCATION, XH0,*
     1           * WAS FOUND IN THE INTERVAL (0,A).*/
     2           * IT IS ASSUMED TO BE THE VALUE OF A.*)
 9904 FORMAT (/1H0,*THE BISECTION METHOD USED TO FIND THE ORIGINAL*
     1           * HINGE LOCATION, XH0, DID NOT CONVERGE AFTER 100*
     2           * ITERATIONS.*/
     3           * THE RESULT OF THE LAST ITERATION WILL BE ASSUMED*
     4           * CORRECT:  XH0 = *, G15.8, *.*)
C
      END
```

```
      SUBROUTINE BISECT (GUESS1, GUESS2, ROOT, ERRFLG, FTN)
C
C          THIS SUBROUTINE TRIES TO FIND A ZERO ROOT FOR THE GIVEN
C          FUNCTION WITHIN THE INTERVAL (GUESS1, GUESS2) USING THE
C          BISECTION METHOD.  GUESS 1 MUST BE LESS THAN GUESS 2.
C          NOTE:  THIS ROUTINE MAY BLOW UP IF THE FUNCTION, FTN, HAS
C          A DISCONTINUITY IN THE GIVEN INTERVAL.
C
C          THE INPUT PARAMETERS ARE DEFINED AS FOLLOWS:
C             GUESS1:   THE LOWER END OF THE INTERVAL BEING CONSIDERED
C             GUESS2:   THE UPPER END OF THE INTERVAL BEING CONSIDERED
C             FTN:      THE EXTERNAL FUNCTION SUBPROCEDURE WHICH COMPUTES
C                       THE FUNCTION FOR WHICH BISECT IS ATTEMPTING TO
C                       TO FIND A ROOT.
C
C          THE OUTPUT PARAMETERS ARE DEFINED AS FOLLOWS:
C             ROOT:     THE ROOT FOR THE FUNCTION, FTN, IN THE GIVEN
C                          INTERVAL , IF FOUND
C                       THE MIDPOINT OF THE INTERVAL, IF THE ROOT WAS
C                          NOT FOUND
C             ERRFLG:   0,    THE ROOT WAS FOUND SUCCESSFULLY
C                       -1,   THE ROOT FAILED TO CONVERGE AFTER 100
C                          ITERATIONS
C                       +1,   THE ROOT COULD NOT BE FOUND AS FTN(GUESS1)
C                          AND FTN(GUESS2) HAVE THE SAME SIGN, INFERRING
C                          THAT THERE IS EITHER NO ROOT OR MORE THAN ONE
C                          ROOT FOR THE FUNCTION IN THIS INTERVAL.
C
C          THIS ROUTINE IS CALLED BY THE SUBPROCEDURES, PTZERO AND
C          CALXH0.  IT CALLS THE SUBPROCEDURE, FTN, WHICH MAY BE ONE
C          OF THE FOLLOWING FUNCTIONS:
C             FTNZ:    CALLED FROM PTZERO, TO FIND Z
C             BFTNX:   CALLED FROM CALXH0, TO FIND XH0 FOR A BEAM CASE
C             PFTNX:   CALLED FROM CALXH0, TO FIND XH0 FOR A PLATE CASE.
C
C
      INTEGER  ERRFLG
      EXTERNAL  FTN
C
C
C          CLEAR ERROR-FLAG AND INITIALIZE END POINT VALUES
C          INITIALIZE EPSILON AND PUT DUMMY VALUE IN F3
C
      ERRFLG = 0
      ITER = 0
      X1 = GUESS1
      X2 = GUESS2
      EPSILN = ABS(X2 - X1) * 0.00000001
      F3 = 1.0
      SF3 = SIGN (1.0, F3)
C
      F1 = FTN (X1)
      F2 = FTN (X2)
      SF1 = SIGN (1.0, F1)
      SF2 = SIGN (1.0, F2)
```

```
C
C          IF THE SIGNS OF F1 AND F2 ARE THE SAME
C
      IF (SF1.NE.SF2)  GO TO 1000
C
C          THEN THERE IS AN EVEN NUMBER OF ROOTS IN THE GIVEN INTERVAL
C            (0,2,4,...)--SET THE ERROR FLAG
C
        ERRFLG = 1
        ROOT = (X2 - X1)/2.0 + X1
        GO TO 5000
C
C          ELSE CONTINUE
C
C          DO WHILE F3 IS NOT ZERO AND ITER < 100
C                    AND (X2 - X1) IS NOT ZERO
C
 1000   IF ((SF3*F3) .LT. EPSILN  .OR.
     1       (X2 - X1) .LT. EPSILN    )  GO TO 4900
          ITER = ITER + 1
          IF (ITER .GT. 100)  GO TO 4800
C
C            FIND X3 BETWEEN X1 AND X2.  COMPUTE F3.
C
          X3 = X1 + (X2 - X1)/2.0
          F3 = FTN (X3)
          SF3 = SIGN (1.0, F3)
C
C            IF NO ROOT BETWEEN X1 AND X3
C
          IF (SF3.NE.SF1)  GO TO 2000
C
C              THEN MOVE X1 TO X3
C
            X1 = X3
            F1 = F3
            SF1 = SF3
            GO TO 3000
C
C              ELSE MOVE X2 TO X3
C
 2000       X2 = X3
            F2 = F3
            SF2 = SF3
C
C            END IF
C
 3000     GO TO 1000
C
C          END WHILE
C
C          TAKE CARE OF ERROR--DID NOT CONVERGE IN 100 TRIES
C
 4800   ERRFLG = -1
        ROOT = (X2 - X1)/2.0 + X1
```

```
      GO TO 5000
C
C        ROOT HAS BEEN FOUND
C
 4900   ROOT = X3
C
C       END IF
C
 5000 RETURN
C
      END
```

```
      FUNCTION FTNZ (TRIALZ)
C
C          THIS FUNCTION CALCULATES A VALUE FOR THE PLATE Z FUNCTION
C          FOR THE GIVEN TRIAL Z.  IF TRIAL Z IS A CORRECT GUESS
C          AT Z, THE RESULT WILL BE ZERO.
C
C          THIS FUNCTION IS CALLED BY THE ROOT-FINDING SUBPROCEDURE,
C          BISECT, WHICH WAS CALLED BY THE SUBPROCEDURE, PTZERO.  IT
C          CALLS NO SUBPROCEDURES OTHER THAN THE SYSTEM LIBRARY
C          ROUTINE, EXP.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
C
C
      TRZSQ = TRIALZ * TRIALZ
C
C          IF THIS IS A LINEAR LOAD
C
      IF (LOADFG.NE.1)  GO TO 1000
C
C             THEN USE LINEAR LOAD FORMULA FOR Z
C
         FTNZ = PE*(4.0*ARSQ*TRIALZ + 8.0*TRIALZ -12.0)
     1          + PC*(2.0*ARSQ*TRIALZ**3 + 3.0*TRZSQ - 5.0)
         GO TO 5000
C
C             ELSE USE BLAST LOAD FORMULA FOR Z
C
 1000    EXPBZ1 = EXP (BETA * (TRIALZ - 1.0) )
         FTNZ = ARSQ*TRIALZ*(PE*BETAFR*TRZSQ*TRIALZ*SIXTH
     1                     + PC*(EXPBZ1*(6.0 - 4.0*BETA*TRIALZ
     2                                 +BETASQ*TRZSQ)
     3                     - EXPBET*(6.0 + 2.0*BETA*TRIALZ) ) )
     4        + PE*BETAFR*TRZSQ*(TRIALZ*THIRD - HALF)
     5        + PC*(EXPBZ1*(3.0 - 3.0*BETA*TRIALZ + BETASQ*TRZSQ)
     6            + EXPBET*(2.0*BETA*TRZSQ + 9.0*TRZSQ - 3.0)
     7            - 2.0*BETASQ*TRZSQ + 7.0*BETA*TRZSQ - 9.0*TRZSQ)
C
C          END IF (TYPE OF LOAD)
C
 5000 RETURN
C
      END
```

```
      FUNCTION BFTNX (TRIALX)
C
C           THIS FUNCTION CALCULATES A VALUE OF THE BEAM ORIGINAL
C           HINGE LOCATION EQUATION FOR THE GIVEN TRIAL X (WHICH IS A
C           GUESS AT THE CORRECT VALUE).  IF THE RESULT RETURNED IS
C           ZERO, THE GUESS IS CORRECT.
C
C           THIS FUNCTION IS CALLED BY THE ROOT-FINDING ROUTINE,
C           BISECT.  IT CALLS NO SUBPROCEDURES OTHER THAN THE LIBRARY
C           FUNCTION, EXP.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BFFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C
      TRXSQ = TRIALX*TRIALX
      ONEMTX = 1.0 - TRIALX
C
C           USE EQUATION DEPENDING ON TYPE OF LOAD FOR BEAM
C
C           IF LINEAR LOAD
      IF (LOADFG.NE.1)  GO TO 1000
C
        BFTNX = ASQ*TRXSQ*(PE - PC*ONEMTX - AIDA*W) - 6.0*F*XMU
C
C
C           IF BLAST LOAD
 1000 IF (LOADFG.NE.2)  GO TO 5000
C
        BFTNX = ONEMTX*BETACB*(ASQ*TRXSQ*(PE - AIDA*W) - 6.0*F*XMU)
     1          + 2.0*ASQ*PC*(EXP(BETA*(-ONEMTX))
     2                       *(3.0*ONEMTX*(BETASQ*TRXSQ + 2.0
     3                                     - 2.0*BETA*TRIALX)
     4                         + BETA*TRXSQ*(BETA*TRIALX - 1.0))
     5                    - 6.0*ONEMTX*EXPBET
     5                    - BETA*TRXSQ*(BETA - 1.0) )
C
C          END OF CASE ON TYPE OF LOAD
C
 5000 RETURN
C
      END
```

```
      FUNCTION PFTNX (TRIALX)
C
C          THIS FUNCTION CALCULATES A VALUE OF THE PLATE ORIGINAL
C          HINGE LOCATION EQUATION FOR THE GIVEN TRIAL X (WHICH IS A
C          GUESS AT THE CORRECT VALUE).  IF THE RESULT RETURNED IS
C          ZERO, THE GUESS IS CORRECT.
C
C          THIS FUNCTION IS CALLED BY THE ROOT-FINDING PROCEDURE,
C          BISECT.  IT CALLS THE SUBPROCEDURE, CALBAR, TO CALCULATE
C          THE ELEMENTS OF DELTA DOT DOT AND THETA DOT DOT AT TIME
C          = 0, FOR THE GIVEN TRIAL X.
C
C
      COMMON /BARS  /  BARA, BARB, BARC, BARD, BARDEL, BARDNM, BARG,
     1                 BARH, BARJ, BARK, BARR, BARS, BART, BARTHE, BARU,
     2                 EXPBX1, EXPBZX
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
C
      CALL CALBAR(TRIALX)
C
      PFTNX = A * TRIALX * (BARTHE + BARC*BARD + BARG*BARH)
     1        -  BARDEL  +  DELTAK
C
      RETURN
C
      END
```

```
      SUBROUTINE TSTEP (LOOPFG)
C
C          THIS SUBROUTINE CONTAINS THE LOOP OF OPERATIONS PERFORMED
C          ON THE GIVEN VARIABLES FOR THE CASE FOR ONE TIME STEP.
C
C          THIS SUBROUTINE IS CALLED BY THE TIME-LOOP-CONTROL
C          PROCEDURE, TCNTRL.  IT CALLS THE FOLLOWING SUBPROCEDURES:
C             CHEKXH:  TO CHECK THE LOCATION OF THE HINGE, XH, DURING
C                      MECHANISM TWO OF A CASE
C             PRINTR:  TO PRINT OUT A LINE OF RESULTS FOR THE GIVEN
C                      TIME STEP
C             RUNGEK:  TO PERFORM THE RUNGE-KUTTA COMPUTATION OF THE
C                      VARIABLES AT THE CURRENT TIME STEP.
C
C          THE PARAMETER, LOOPFG, IS THE LOOP-CONTROL-FLAG, WHICH IS SET
C          BY THIS ROUTINE IF THE LOOP SHOULD BE TERMINATED FOR ANY
C          REASON.
C
C          THE CASE-IS-DONE-FLAG, DONEFG, CAN BE SET IN TWO WAYS:
C            1.:  IF EITHER VELOCITY, DELTA DOT OR THETA DOT, GO NEGATIVE
C                 INDICATING THAT THE MAXIMUM DEFLECTION HAS BEEN REACHED
C                 OR THE PRESSURE WAS NOT SUFFICENT TO SHOW A RESPONSE
C                 AND THE CASE CAN BE TERMINATED
C            2.:  IF THE TIME MAXIMUM (TMAX) FOR THE CASE TO BE RUN HAS
C                 BEEN REACHED, IN WHICH CASE THE CASE MUST BE ENDED.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
      DATA   STAR /1H*/
C
C
C          COMPUTE EQUATIONS OF MOTION AND WORK FOR THE NEXT TIME STEP
C
      CALL RUNGEK
      WK     = WF - WP - V
      T      = T + TINCR
      STEPCT = STEPCT + TINCR
C
C          IF MECHANISM 1
C
      IF (MECHFG.NE.1)  GO TO 1100
```

```
C
C             THEN COMPLETE VEL AND DELTA CALCULATIONS
C
        VEL = THETAD * A
        DELTA = THETA * A
C
C             ELSE CONTINUE
C           END IF (MECH 1)
C
C           IF EITHER VELOCITY IS ZERO
C
 1100 IF (VEL.GE.0.0 .AND. THETAD.GE.0.0)  GO TO 1300
C
C       THEN WRITE MESSAGE AND SET
C            CASE-IS-DONE-FLAG TO TERMINATE CASE
C
C                   IF THIS IS THE FIRST TIME STEP
        IF (T.NE.TINCR)  GO TO 1150
C                         THEN WRITE NO RESPONSE MESSAGE
          WRITE (6,9903)
          GO TO 1200
C                         ELSE WRITE MAXIMUM DEFLECTION FOUND
 1150     WRITE (6,9901)  DELTA, T
C
C              END IF (1ST TIME STEP)
C
 1200   LOOPFG = 1
        DONEFG = 1
        WRITE (6,9904)
        GO TO 2000
C
C         ELSE CONTINUE COMPUTATIONS
C
C          IF MECHANISM 2, GET NEW HINGE LOCATION (XH)
 1300   IF (MECHFG.EQ.2)  CALL CHEKXH
C
C            IF HINGE LOCATION IS OKAY
        IF (BADXFG.NE.0)  GO TO 1500
C
C              THEN CHECK FOR FAILURE WITH THETA U
C                AND PRINT RESULTS IF END OF TIME STEP INTERVAL
C
          THETAU = D * (SQRT(THETU1*XH + 1.0) - 1.0) / XH
          IF (THETA.GT.THETAU)  FLAG = STAR
          IF ((STEPCT + TINCR).GT.TPRINT)  CALL PRINTR
C
C               IF TIME HAS REACHED THE LIMIT (TMAX)
C
          IF (TMAX.GE.(T + TINCR))  GO TO 1400
C
C                THEN SET LOOP-CONTROL- AND CASE-IS-DONE-FLAGS
C                 AND WRITE TIME-EXCEEDED MESSAGE
            LOOPFG = 1
            DONEFG = 1
            WRITE (6,9902)
```

```
          WRITE (6,9904)
C               ELSE CONTINUE
C              END IF (T=TMAX)
C
 1400      GO TO 1900
C
C              ELSE SET LOOP CONTROL FLAG TO TERMINATE THIS TRY
C                   BECAUSE OF A BAD HINGE LOCATION
 1500      LOOPFG = 1
C
C              END IF (GOOD XH)
C
 1900    GO TO 2000
C
C           END IF (ZERO VELOCITIES)
C
 2000 RETURN
C
C          FORMAT STATEMENTS
C
 9901 FORMAT (1H0, 20X, *MAXIMUM DEFLECTION = *, G15.8, * AT TIME = *,
    1          G15.8)
 9902 FORMAT (1H0, 30X, *TIME EXCEEDED*)
 9903 FORMAT (1H0, 20X, *INSUFFICIENT PRESSURE TO GIVE A RESPONSE*)
 9904 FORMAT (1H0, 20X, *AN ASTERISK INDICATES THAT A REINFORCING *,
    1          *ELEMENT HAS FRACTURED*)
C
      END
```

```
      SUBROUTINE RUNGEK
C
C          THIS SUBROUTINE SOLVES FIRST AND SECOND ORDER SIMULTANEOUS
C          DIFFERENTIAL EQUATIONS USING THE RUNGE-KUTTA FOURTH-ORDER
C          METHOD.
C
C          THE FIRST ORDER SOLUTIONS OF THE DIFFERENTIAL EQUATIONS OF
C          THE HIGHEST ORDER (WHICH CAN BE FIRST OR SECOND) ARE STORED
C          IN THE ARRAY CALLED Y.  THE FINAL SOLUTIONS OF THE DIFFER-
C          ENTIAL EQUATIONS (WHICH ARE ACTUALLY THE FIRST ORDER RESULTS
C          STORED IN THE ARRAY, Y) ARE STORED IN THE ARRAY U, WITH
C          MATCHING SUBSCRIPT.
C
C          THE ARRAY, ARG, HOLDS THE VALUES TO BE USED IN EACH STEP
C          OF COMPUTING THE FUNCTION.  THE FIRST ELEMENT OF ARG, ARG(1),
C          IS ALWAYS USED FOR THE TIME ARGUMENT, T.
C
C          THE ARRAYS, DX AND DXDX, ARE USED FOR THE VALUES OF THE FIRST
C          AND SECOND ORDER CHANGE (DELTA) DETERMINED FOR EACH VARIABLE
C          BY THE RUNGE-KUTTA METHOD.  THESE ARE ADDED TO EACH ELEMENT
C          OF Y AND U, RESPECTIVELY, TO DETERMINE THE NEW VALUES OF
C          THE VARIABLES AT THE END OF THE TIME STEP.
C
C          THIS SUBROUTINE CALLS THE SUBPROCEDURE, COMP, WHICH
C          ACTUALLY COMPUTES THE PROPER FUNCTIONS AT EACH POINT OF THE
C          RUNGE-KUTTA CALCULATION.
C
C          THE ARRAYS, A0, A1, A2, AND A3, HOLD THE INTERMEDIATE
C          RESULTS OF THE RUNGE-KUTTA CALCULATION, SHOWING THE VALUE
C          OF EACH FUNCTION AT EACH POINT OF THE CALCULATION.  THESE
C          VALUES ARE COMBINED TO FORM THE VALUES OF THE ARRAYS, DX
C          AND DXDX, ACCORDING TO THE RUNGE-KUTTA FORMULA.
C
C          FOR FURTHER INFORMATION ON THE METHOD USED HERE, THE READER
C          IS DIRECTED TO THE TEXT, INTRODUCTION TO NUMERICAL ANALYSIS
C          BY HILDEBRAND (NEW YORK: MCGRAW-HILL, 1956), PP. 233-239.
C
C          THIS SUBROUTINE IS CALLED BY THE PROCEDURE, TSTEP.  IT
C          CALLS THE SUBPROCEDURE, COMP.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
C
C
      DIMENSION  Y(5),  U(2),
     1           DX(5),  DXDX(2),
     2           A0(5), A1(5), A2(5), A3(5),
     3           ARG(5)
```

```
C
C
C       INITIALIZE VALUES OF FIRST AND SECOND ORDER EQUATIONS
C
      Y(1) = VEL
      U(1) = DELTA
      Y(2) = THETAD
      U(2) = THETA
      Y(3) = WF
      Y(4) = WP
      Y(5) = V
C
C         SET UP ARGUMENTS FOR FIRST STEP OF RUNGE-KUTTA
C
      ARG(1) = T
      ARG(2) = DELTA
      ARG(3) = VEL
      ARG(4) = THETA
      ARG(5) = THETAD
      CALL COMP (ARG, A0)
C
C         SET UP ARGUMENTS FOR THE SECOND STEP
C
      HALFTI = 0.5*TINCR
      ARG(1) = T + HALFTI
      ARG(2) = DELTA + HALFTI*VEL
      ARG(3) = VEL + 0.5*A0(1)
      ARG(4) = THETA + HALFTI*THETAD
      ARG(5) = THETAD + 0.5*A0(2)
      CALL COMP (ARG, A1)
C
C         SET UP ARGUMENTS FOR THE THIRD STEP
C
      ARG(2) = DELTA + HALFTI*VEL + 0.5*HALFTI*A0(1)
      ARG(3) = VEL + 0.5*A1(1)
      ARG(4) = THETA + HALFTI*THETAD + 0.5*HALFTI*A0(2)
      ARG(5) = THETAD + 0.5*A1(2)
      CALL COMP (ARG, A2)
C
C         SET UP ARGUMENTS FOR THE FOURTH AND LAST STEP
C
      ARG(1) = T + TINCR
      ARG(2) = DELTA + TINCR*VEL + HALFTI*A1(1)
      ARG(3) = VEL + A2(1)
      ARG(4) = THETA + TINCR*THETAD + HALFTI*A1(2)
      ARG(5) = THETAD + A2(2)
      CALL COMP (ARG, A3)
C
C         PUT PIECES TOGETHER
C
      DO 1000  I=1,2
      DXDX(I) = TINCR*Y(I) + TINCR*(A0(I)+A1(I)+A2(I))/6.0
 1000 U(I) = U(I) + DXDX(I)
C
      DO 1100  I = 1,5
```

```
      DX(I) = (A0(I) + 2.0*A1(I) + 2.0*A2(I) + A3(I))/6.0
 1100 Y(I) = Y(I) + DX(I)
C
C         COMPUTATION COMPLETE
C
      VEL = Y(1)
      DELTA = U(1)
      THETAD = Y(2)
      THETA = U(2)
      WF = Y(3)
      WP = Y(4)
      V = Y(5)
C
      RETURN
      END
```

```
      SUBROUTINE COMP (ARG, AA)
C
C         THIS SUBROUTINE ACTUALLY COMPUTES ONE OF EIGHT POSSIBLE
C         SETS OF FUNCTIONS WHICH DESCRIBE THE SIMULTANEOUS DIFFER-
C         ENTIAL EQUATIONS BEING SOLVED FOR EACH TIME STEP.  THE
C         SET OF EQUATIONS USED DEPENDS ON THE TYPE OF CASE AND
C         TYPE OF LOAD AS WELL AS THE CURRENT STATE (OR MECHANISM)
C         OF THE CASE.
C
C         THE INPUT AND OUTPUT PARAMETER ARRAYS ARE DEFINED AS
C         FOLLOWS:
C           ARG:  INPUT TO THE PROCEDURE, CONTAINS THE ARGUMENTS TO
C                 BE USED IN COMPUTING THE FUNCTIONS
C           AA:   OUTPUT RESULTS TO BE RETURNED TO THE CALLING
C                 PROCEDURE, RUNGEK.
C
C         THIS SUBROUTINE CALLS THE SUBPROCEDURE, CALBAR, TO COMPUTE
C         PORTIONS OF THE PLATE EQUATIONS OF MOTION.  IT ALSO USES THE
C         SYSTEM LIBRARY ROUTINE, EXP.  THIS ROUTINE IS CALLED BY THE
C         RUNGE-KUTTA SUBPROCEDURE, RUNGEK.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /BARS  /  BARA, BARB, BARC, BARD, BARDEL, BARDNM, BARG,
     1                 BARH, BARJ, BARK, BARR, BARS, BART, BARTHE, BARU,
     2                 EXPBX1, EXPBZX
      COMMON /COMPS /  ATHEDB, ATHEDL, ATHED1, ATHED2, AVDOT, AWFDB,
     1                 AWFDL, BTHED2, BTHED3, BVDOT, BWFDOT, BWPDOT,
     2                 PWFDB, PWFDL, PWPDOT, PVDOT
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
C
      DIMENSION  ARG(5),  AA(5)
C
C
C         DETERMINE FTNT
C            USING CASE ON RATIO OF T:TAU WITH THE WAVE-FUNCTION
C
C
      FTNT = 0.0
      TRATIO = ARG(1) / TAU
      IF (TRATIO.LE.1.0 .AND. WAVEFN.EQ.1.0)
     1     FTNT = (1.0 - TRATIO) * EXP(-ALPHA * TRATIO)
      IF (TRATIO.LE.1.0 .AND. WAVEFN.EQ.2.0)
     1     FTNT = 1.0
```

```
C
C
C          IF FIRST TIME OR SECOND TIME THROUGH THIS ROUTINE
C
      IF (M1STFG.EQ.0)  GO TO 900
C
C            THEN, TO AVOID ZERO DIVISION AND TO INITIALIZE THE FIRST
C            TIME STEP FOR THE RUNGE-KUTTA METHOD, SET THE SECOND AND
C            FOURTH ARGUMENTS SUCH THAT CURRX WILL REFLECT THE VALUE
C            OF THE ORIGINAL HINGE LOCATION.  THIS WILL BE DONE THE
C            FIRST AND SECOND TIME THE PROCEDURE COMP IS CALLED:
C            THAT IS, FOR THE CALCULATION OF THE FIRST TWO PARAMETERS
C            OF THE RUNGE-KUTTA FOR THE FIRST TIME STEP ONLY.
C
        ARG(2) = XH
        ARG(4) = 1.0
        M1STFG = M1STFG - 1
C
C            ELSE CONTINUE
C          END IF (1ST OR SECOND TIME THRU)
C
C
C          CASE ON COMBINATION OF LOAD-, MECHANISM-, AND BEAM-PLATE-FLAGS
C
  900 LMBPSM = (LOADFG - 1)*4 + (BPFLAG - 1)*2 + (MECHFG - 1)*1 + 1
      GO TO (1000, 2000, 3000, 4000,
     1       5000, 6000, 7000, 8000), LMBPSM
C
C          LINEAR LOAD CASES
C
C            THIS IS A BEAM CASE IN MECHANISM 1, LINEAR LOAD
C
 1000   CURRX = 1.0
        AA(1) = 0.0
        AA(2) = FTNT * ATHEDL  -  ATHED2
        AA(3) = FTNT * ARG(5) * AWFDL
        AA(4) = BWPDOT * ARG(5)
        AA(5) = AVDOT * ARG(5)
        GO TO 9000
C
C            THIS IS A BEAM CASE IN MECHANISM 2, LINEAR LOAD
C
 2000   CURRX = ARG(2) / (A * ARG(4))
        AA(1) = FTNT*(PE + PC*HALF*(CURRX + 1.0) )/XM  -  DELTAK
        AA(2) = FTNT * (3.0*PE + 2.0*CURRX*PC)
     1              / (2.0 * A * CURRX * XM)
     2         - BTHED2/CURRX  -  BTHED3/(CURRX**3)
        AA(3) = FTNT * ARG(5) * BWFDCT * CURRX
     1         * (PE * (1.0 - CURRX*HALF)
     2            + PC * HALF * (1.0 - THIRD*CURRX**2) )
        AA(4) = BWPDOT * ARG(5)
        AA(5) = BVDOT * ARG(5) * CURRX * (1.0 - CURRX*HALF)
        GO TO 9000
C
C            THIS IS A PLATE CASE IN MECHANISM 1, LINEAR LOAD
```

```
C
 3000   CURRX = 1.0
        AA(1) = 0.0
        AA(2) = FTNT * ATHED1  +  ATHED2
        AA(3) = FTNT * ARG(5) * AWFDL
        AA(4) = PWPDOT * ARG(5)
        AA(5) = AVDOT * ARG(5)
        GO TO 9000
C
C          THIS IS A PLATE CASE IN MECHANISM 2, LINEAR LOAD
C
 4000   CURRX = ARG(2) / (A * ARG(4))
        CALL CALBAR(CURRX)
        AA(1) = FTNT * BARDEL  -  DELTAK
        AA(2) = FTNT * BARTHE  +  BARC * BARD  +  BARG * BARH
        AA(3) = FTNT * ARG(5) * PWFDL
     1            * (PE*CURRX*(1.0 - CURRX*HALF*ONEPZ + Z*THIRD*CURRX**2)
     2             + PC*CURRX*(THIRD - ONEPZ*SIXTH*CURRX*CURRX
     3                        + (SIXTH - Z*ONEMZ*FOURTH)*CURRX**3) )
        AA(4) = PWPDOT * ARG(5)
        AA(5) = PVDOT * ARG(5) * CURRX
     1             * (1.0 - ONEPZ*HALF*CURRX + Z*THIRD*CURRX*CURRX)
        GO TO 9000
C
C        BLAST LOAD CASES
C
C          THIS IS A BEAM CASE IN MECHANISM 1, BLAST LOAD
C
 5000   CURRX = 1.0
        AA(1) = 0.0
        AA(2) = FTNT * ATHEDB  -  ATHED2
        AA(3) = FTNT * ARG(5) * AWFDB
        AA(4) = BWPDOT * ARG(5)
        AA(5) = AVDOT * ARG(5)
        GO TO 9000
C
C          THIS IS A BEAM CASE IN MECHANISM 2, BLAST LOAD
C
 6000   CURRX = ARG(2) / (A * ARG(4))
        BETAX = BETA * CURRX
        ONEMCX = 1.0 - CURRX
        EXPBCX = EXP (BETA * (-ONEMCX) )
        AA(1) = (FTNT/XM) * (PE
     1                       + (PC/(BETASQ*ONEMCX))
     2                       *(BETA - 1.0 - (BETAX - 1.0)*EXPBCX) )
     3          - DELTAK
        AA(2) = (3.0*FTNT / (XM*A*CURRX**3) )
     1            * (PE*HALF*CURRX*CURRX
     2             + (PC/BETACB) * (EXPBCX*(BETAX*(BETAX - 2.0) + 2.0)
     3                              - 2.0*EXPBET) )
     4          -  BTHED2/CURRX  -  BTHED3/(CURRX**3)
        AA(3) = FTNT * ARG(5) * BWFDOT
     1          * (PE * CURRX *(1.0 - CURRX*HALF)
     2             + (PC/BETACB) * (EXPBCX*(2.0 - BETAX)
     3                          + BETAX*(BETA - 1.0) - 2.0*EXPBET) )
```

```
      AA(4) = BWPDOT * ARG(5)
      AA(5) = BVDOT * ARG(5) * CURRX * (1.0 - CURRX*HALF)
      GO TO 9000
C
C          THIS IS A PLATE CASE IN MECHANISM 1, BLAST LOAD
C
 7000 CURRX = 1.0
      AA(1) = 0.0
      AA(2) = FTNT * ATHED1  +  ATHED2
      AA(3) = FTNT * ARG(5) * AWFDB
      AA(4) = PWPDOT * ARG(5)
      AA(5) = AVDOT * ARG(5)
      GO TO 9000
C
C          THIS IS A PLATE CASE IN MECHANISM 2, BLAST LOAD
C
 8000 CURRX = ARG(2) / (A * ARG(4))
      CALL CALBAR(CURRX)
      AA(1) = FTNT * BARDEL  -  DELTAK
      AA(2) = FTNT * BARTHE   +  BARC * BARD  +  BARG * BARH
      AA(3) = FTNT * ARG(5) * PWFDB
     1       * (PE*BETAFR*ZSQ*(CURRX - Z*CURRX*CURRX*HALF
     2                          + ONEPZ*THIRD*CURRX**3)
     3      +  PC*(2.0*BETASQ*ZSQ*CURRX  -  4.0*BETA*ZSQ*CURRX
     4            +  EXPBZX*(BETASQ*ZSQ*CURRX*(CURRX - 1.0)
     5                      - 3.0*BETA*Z*CURRX + 2.0*BETA*Z + 3.0)
     6            +  EXPBX1*ZSQ*(BETASQ*CURRX*CURRX - 5.0*BETA*CURRX
     7                         - BETASQ*CURRX + 2.0*BETA + 9.0)
     8            -  EXPBET*(2.0*BETA*Z*ONEPZ + 9.0*ZSQ + 3.0) ) )
      AA(4) = PWPDOT * ARG(5)
      AA(5) = PVDOT * ARG(5) * CURRX
     1         * (1.0 - ONEPZ*HALF*CURRX + Z*THIRD*CURRX*CURRX)
C
C         END CASE ON LOADFG, BPFLAG & MECHFG
C
C
C        MULTIPLY EACH TERM BY THE TIME STEP INCREMENT
C
 9000 DO 9100  I=1,5
 9100 AA(I) = TINCR * AA(I)
C
      RETURN
      END
```

```
      SUBROUTINE CALBAR (XX)
C
C          THIS SUBROUTINE CALCULATES PIECES OF THE PLATE EQUATIONS
C          OF MOTION, THE BARRED ELEMENTS MAKING UP DELTA DOT DOT AND
C          THETA DOT DOT.
C
C          THIS PROCEDURE IS CALLED BY THE FOLLOWING SUBPROCEDURES:
C            PTZERO:  TO GET ELEMENTS OF THE EQUATIONS OF MOTION TO
C                     DEFINE MECHANISM 1 CONSTANTS FOR THE PROCEDURE,
C                     COMP
C            COMP:    TO COMPUTE ELEMENTS OF THE EQUATIONS OF MOTION
C                     AT THE GIVEN TIME STEP FOR THE RUNGE-KUTTA
C                     ROUTINE, RUNGEK
C            PFTNX:   TO COMPUTE ELEMENTS OF THE EQUATIONS OF MOTION
C                     MAKING UP THE ORIGINAL HINGE LOCATION FUNCTION
C                     AT TIME ZERO IN ORDER TO FIND A ROOT FOR SAID
C                     FUNCTION.
C          IT CALLS NO SUBPROCEDURE OTHER THAN THE LIBRARY SYSTEM
C          EXPONENTIAL FUNCTION, EXP.
C
C          ALL INPUT COMES THROUGH THE COMMON BLOCK, CONSTS, WITH
C          THE EXCEPTION OF THE SINGLE PARAMETER, XX, WHICH IS DEFINED
C          AS FOLLOWS:
C            1:  THE CURRENT VALUE OF THE HINGE LOCATION (FROM COMP)
C            2:  THE CURRENT GUESS AT THE ORIGINAL HINGE LOCATION
C                (FROM PFTNX)
C            3:  THE FINAL VALUE OF THE HINGE LOCATION (FROM PTZERO).
C          ALL OUTPUT IS PASSED THROUGH THE COMMON BLOCK, BARS.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /BARS  /  BARA, BARB, BARC, BARD, BARDEL, BARDNM, BARG,
     1                 BARH, BARJ, BARK, BARR, BARS, BART, BARTHE, BARU,
     2                 EXPBX1, EXPBZX
      COMMON /CONSTS/  ACUBE, ARSQ, ARSZP1, ARZSP1, ASQ, B, BARA1,
     1                 BARC1, BARG1, BARH1, BARJ1, BETACB, BETAFR,
     2                 BETASQ, DELTAK, EPSLNU, EXPBET, FOURTH, HALF,
     3                 ONEMZ, ONEPZ, SIXTH, THETU1, THIRD, W, XMU, Z,
     4                 ZCUBE, ZFOUR, ZSQ
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE,
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
C
C
      XXSQ = XX*XX
      XXCUBE = XX**3
      ONEMXX = 1.0 - XX
      ONEMZX = 1.0 - Z*XX
C
      EXPBX1 = EXP(BETA*(- ONEMXX))
      EXPBZX = EXP(BETA*(- ONEMZX) )
C
      BARDNM = ARZSP1*(THIRD - XX*FOURTH) + ONEMZ*XX*FOURTH
```

```
C
      BARC = BARC1 / XX
      BARD = (ARZSP1*(HALF - XX*THIRD) + ONEMZ*XX*THIRD) / BARDNM
      BARG = BARG1 / XXCUBE
      BARH = BARH1 / BARDNM
C
C
C         CASE ON TYPE OF LOAD
C
C            LINEAR LOAD
      IF (LOADFG.NE.1)  GO TO 2000
C
         BARR = 1.0 / (XM * A * XX * BARDNM)
         BARS = PE*(ARZSP1*(HALF - XX*THIRD) + ONEMZ*XX*THIRD)
     1           + PC*(ZCUBE*XX*AR*(THIRD - XX*FOURTH)
     2                + XXSQ*FOURTH*(11.0*SIXTH - ZSQ*HALF - XX) )
         BARTHE = BARR * BARS
C
C             IF HINGE NOT IN CENTER OF PLATE
C
         IF (XX.EQ.1.0)  GO TO 2000
C
C                THEN COMPUTE DELTA DOT DOT ELEMENTS
C
          BART = 1.0 / (XM * ONEMZX)
          BARU = PE*ONEMZX + PC*(THIRD*(1.0 + XX)
     1                          - XXSQ*HALF*(ZSQ + THIRD) )
          BARDEL = BART * BARU
C
C                ELSE CONTINUE
C               END IF (HINGE IN CENTER)
C
C            IF BLAST LOAD
 2000 IF (LOADFG.NE.2)  GO TO 5000
C
         BARA = BARA1 / BARDNM
         BARB = PE*BETAFR*ZSQ*XXSQ*
     1               (AR*ZSQ*(HALF - XX*THIRD) + ONEMZ*THIRD*XX + 1.0)
     2           + PC*(-EXPBET*(2.0*BETA*ZSQ*(AR + 1.0) + 9.0*ZSQ
     3                       + 6.0*AR*Z - 3.0)
     4              + EXPBX1*(BETACB*ZSQ*XXSQ*ONEMXX
     5                       + BETASQ*ZSQ*(4.0*XXSQ - 2.0*XX)
     6                       - 9.0*BETA*ZSQ*XX + 2.0*BETA*ZSQ + 9.0*ZSQ)
     7              + EXPBZX*(AR*(BETACB*ZFOUR*XXSQ*ONEMXX
     8                        + BETASQ*ZCUBE*(3.0*XXSQ - 2.0*XX)
     9                        - 6.0*BETA*ZSQ*XX + 2.0*BETA*ZSQ + 6.0*Z)
     A                     -  BETASQ*ZSQ*XXSQ
     B                     +  3.0*BETA*Z*XX  -  3.0) )
         BARTHE = BARA * BARB
C
C             IF HINGE NOT IN THE CENTER OF THE PLATE
C
         IF (XX.EQ.1.0)  GO TO 5000
C
C                THEN COMPUTE DELTA DOT DOT ELEMENTS
```

```
C
          BARJ = BARJ1 / (ONEMXX * ONEMZX)
          BARK = BETACB*PE*(Z*XXSQ - ONEPZ*XX + 1.0)
     1             + PC*(2.0*BETA  -  4.0
     2                  + EXPBZX*BETA*ONEMXX*(1.0 - BETA*Z*XX)
     3                  + EXPBX1*(4.0 + BETA - 3.0*BETA*XX
     4                          - BETASQ*XX*ONEMXX) )
          BARDEL = BARJ * BARK
C
C                ELSE CONTINUE
C              END IF (HINGE IN CENTER)
C
C            END CASE ON TYPE OF LOAD
C
C
 5000 RETURN
      END
```

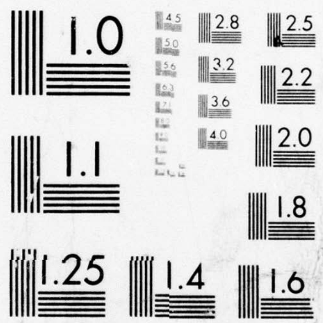
```
      SUBROUTINE CHEKXH
C
C           THIS SUBROUTINE CHECKS THE VALUE OF THE HINGE LOCATION, XH,
C           TO SEE THAT IT IS WITHIN RANGE, AND TO SEE IF IT HAS MOVED
C           FROM A MECHANISM 2 TO A MECHANISM 1 POSITION.
C
C           IF THE HINGE LOCATION IS NEGATIVE (OFF THE END OF THE BEAM OR
C           PLATE), THE DATA MUST BE BAD.  IF THE HINGE LOCATION HAS MOVED
C           WITHIN TWO PERCENT OF THE CENTER OF THE BEAM OR PLATE, IT IS
C           CONSIDERED TO BE AT THE CENTER, WHICH IS THE MECHANISM 1
C           POSITION, AND THE MECHANISM FLAG IS CHANGED TO REFLECT THIS.
C           IF THE HINGE LOCATION HAS MOVED PAST THE CENTER OF THE BEAM
C           OR PLATE BY MORE THAN TWO PERCENT, THE COMPUTATION IS CON-
C           SIDERED TOO INACCURATE, AND SO THE TIME INCREMENT, TINCR,
C           IS HALVED, A MESSAGE IS PRINTED, AND FLAGS ARE SET TO SHOW
C           THAT THIS HAS HAPPENED AND TO RUN THROUGH THE COMPUTATION
C           AGAIN WITH THE SMALLER TIME STEP.  THE MAXIMUM NUMBER OF
C           TIMES WHICH THIS CAN BE ATTEMPTED IS TWO (AND IS STORED IN
C           THE CONSTANT, MAXTRI, IN THE SUBROUTINE, TCNTRL).
C
C           THE BAD-HINGE-LOCATION-FLAG, BADXFG, IS SET IF THE HINGE
C           LOCATION FOUND IS NEGATIVE OR PAST THE CENTER POINT BY
C           MORE THAN TWO PERCENT.
C
C           THE CASE-IS-DONE-FLAG, DONEFG, IS SET IF THE HINGE LOCATION
C           IS NEGATIVE.
C
C           THE MECHANISM-FLAG, MECHFG, IS CHANGED FROM 2 TO 1 IF THE
C           HINGE LOCATION HAS MOVED TO THE CENTER OF THE BEAM OF PLATE.
C
C           THIS SUBROUTINE IS CALLED BY THE FOLLOWING PROCEDURES:
C             TCNTRL:  TO CHECK THE ORIGINAL HINGE LOCATION AT T=0
C             TSTEP:   TO COMPUTE AND CHECK THE HINGE LOCATION AT
C                      EACH SUCCESSIVE TIME STEP.
C           THIS ROUTINE CALLS NO SUBPROCEDURES.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /INPUTS/  A, AIDA, ALPHA, AR, BETA, D, F, H, PC, PE.
     1                 Q, SIGMAC, SIGMAR, TAU, TINCR, TMAX, TPRINT,
     2                 XM, WAVEFN
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
C
C
C          IF THIS IS NOT THE FIRST TIME STEP (T/=0)
C
      IF (T.EQ.0.0 .OR. THETA.EQ.0.0)  GO TO 100
C
C            THEN WE NEED TO COMPUTE HINGE LOCATION FROM LAST RESULTS
        XH = DELTA / THETA
        X = XH / A
C
```

AD-A069 896 FLORIDA UNIV EGLIN AFB GRADUATE ENGINEERING CENTER F/G 13/13
FAILURE OF UNDERGROUND HARDENED STRUCTURES SUBJECTED TO BLAST L--ETC(U)
APR 79 C A ROSS, C C SCHAUBLE AFOSR-78-3592
UNCLASSIFIED AFOSR-TR-79-0679 NL

3 OF 3

AD
A069896

END
DATE
FILMED
7--79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

```
C           ELSE CONTINUE
C         END IF (T)
C
C         IF THE HINGE LOCATION HAS NOT REACHED THE FINAL HINGE LOC
  100 IF (X.GT.0.98)  GO TO 3000
C
C           THEN CASE IS STILL WITHIN MECHANISM 2
C           IF HINGE LOCATION IS NEGATIVE
C
          IF (XH.GT.0.0)  GO TO 2000
C
C           THEN DATA MUST BE BAD:  WRITE ERROR MESSAGE AND
C              SET BAD-HINGE- AND CASE-IS-DONE- FLAGS TO TERMINATE CASE
C
          WRITE (6,9901)
          BADXFG = 1
          DONEFG = 1
C
C             ELSE CONTINUE--HINGE LOCATION IS WITHIN MECH 2
C
C           END IF (NEG XH)
C
 2000    GO TO 5000
C
C           ELSE HINGE LOCATION HAS MOVED TO MECHANISM 1 LOCATION
C           NEED TO CHECK RANGE
C
C           IF HINGE LOCATION IS WITHIN 2 % OF THE FINAL HINGE LOCATION
 3000    IF (X.GT.1.02)  GO TO 4000
C
C             THEN SET HINGE TO FINAL HINGE LOC AND SET FLAGS TO SHOW
C             MECHANISM 1 HAS BEEN SUCCESSFULLY REACHED
C
          XH = A
          X = 1.0
          MECHFG = 1
          GO TO 4999
C
C             ELSE HINGE LOCATION HAS OVERSHOT THE FINAL HINGE LOC
C             NEED TO DECREASE TIME STEP AND TRY AGAIN FROM T=0
C
 4000    TINCR = TINCR/2.0
         BADXFG = 1
         WRITE (6,9900) A, XH
C
C           END IF (2% OF FINAL HINGE LOC)
C
 4999 GO TO 5000
C
C         END IF (XH REACHED FINAL HINGE LOC)
C
 5000 RETURN
C
C         FORMAT STATEMENTS
C
```

```
9900 FORMAT (1H0, *HINGE LOCATION HAS OVERSHOT FINAL HINGE LOC*/
    1          *    FINAL HINGE LOC = *, G15.8, *   HINGE IS AT *, G15.8/
    2          * TIME INCREMENT HAS BEEN HALVED--CASE WILL BE RERUN*)
 9901 FORMAT (1H0, *HINGE LOCATION IS NEGATIVE--CASE IS TERMINATED*/
    1          * CHECK INPUT DATA VALUES*)
C
      END
```

```
      SUBROUTINE PRINTR
C
C          THIS SUBROUTINE PRINTS THE LINE OF RESULTS FOR THE TIME, T.
C          IT IS CALLED BY THE SUBPROCEDURES, TZERO AND TSTEP.  IT
C          CALLS SUBPROCEDURE PAGE TO HEAD A NEW PAGE, IF THE
C          CURRENT LINE COUNT FOR THE PRINTED PAGE EXCEEDS THE
C          MAXIMUM NUMBER OF LINES PER PAGE.
C
C
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
      COMMON /RESULT/  DELTA, T, THETA, THETAD, V, VEL, WF, WK, WP,
     1                 X, XH
C
C
C          IF LINE COUNT EXCEEDS MAX/PAGE
C
C           THEN HEAD A NEW PAGE
C
      IF (NUMLIN.GE.MAXLIN)  CALL PAGE
C
C           ELSE CONTINUE
C          END IF (LINECOUNT)
C
C          PRINT LINE OF RESULTS, ADD 1 TO LINE COUNT,
C            AND ZERO STEP COUNT
C
      WRITE (6,9900)  T, THETA, FLAG, VEL, DELTA, WF, WP, WK, XH
      NUMLIN = NUMLIN + 1
      STEPCT = 0.0
C
      RETURN
C
C          FORMAT STATEMENT
C
 9900 FORMAT (1X, G14.8, 2X, G14.8, A1, 6(2X, G14.8) )
C
      END
```

```
      SUBROUTINE PAGE
C
C         THIS SUBROUTINE PUSHES THE OUTPUT TO A NEW PAGE AND PROVIDES
C         ALL HEADINGS, INCLUDING CURRENT DATE AND TIME.  IT IS CALLED
C         BY THE SUBPROCEDURE, RWDATA, TO PRINT HEADINGS FOR THE FIRST
C         PAGE OF THE NEW CASE AND BY THE SUBPROCEDURE, PRINTR, FOR
C         EACH ADDITIONAL PAGE OF INPUT THAT FOLLOWS FOR THAT CASE.
C         IT CALLS NO SUBPROCEDURES.
C
C
      INTEGER  BADXFG, BPFLAG, DONEFG, EOFLAG
      COMMON /FLAGS /  BADXFG, BPFLAG, DONEFG, EOFLAG, IERRFG, KOUNT,
     1                 LOADFG, MECHFG, M1STFG, NCONFG
      COMMON /PRINTS/  FLAG, MAXLIN, NAIDA, NF, NUMLIN, NUMPAG, NWAVEF,
     1                 STEPCT, TIMNOW, TITLE(15), TODAY, TYPE
C
      DIMENSION  TYPSUP(2),  TYPWAV(2,2),  TYPSLB(4,3),  TYPLOA(2)
      DATA  TYPSUP / 7H SIMPLY, 7HCLAMPED/,
     1      TYPWAV / 6HGENERA, 6HL TIME,  6H SQUAR, 6HE WAVE/,
     2      TYPSLB / 8HHORIZONT, 8HAL SLAB,, 8H EXPLOSI, 8HVE ABOVE,
     3               8H        , 8HVERTICAL, 8H WALL   , 8H        ,
     4               8HHORIZONT, 8HAL SLAB,, 8H EXPLOSI, 8HVE BELOW/,
     5      TYPLOA / 6HLINEAR, 6H BLAST/
C
C
C         IF WE ARE COMPLETING THE FIRST PAGE FOR CASE
C
      IF (NUMPAG.NE.1)  GO TO 100
C
C           THEN PRINT SUPPORT, WAVE, WEIGHT TYPES AT BOTTOM OF PAGE
C
        WRITE (6,9930)  TYPSUP(NF), (TYPWAV(I,NWAVEF), I=1,2),
     1                  (TYPSLB(I,NAIDA), I=1,4),  TYPLOA(LOADFG)
C
C           ELSE CONTINUE
C          END IF (END OF FIRST PAGE)
C
C         IF WE ARE COMPLETING THE SECOND OR SUCCEEDING PAGE
C
  100 IF (NUMPAG.LT.2)  GO TO 500
C
C           THEN PRINT * MESSAGE AT BOTTOM OF PAGE
C
        WRITE (6,9940)
C
C           ELSE CONTINUE
C          END IF (END OF 2ND PAGE)
C
C         ADD ONE TO PAGE NUMBER  AND
C            PRINT TOP TWO LINES ON NEXT PAGE
C
  500 NUMPAG = NUMPAG + 1
      WRITE (6,9900)  TYPE, TODAY, TIMNOW, TITLE, NUMPAG
C
C         IF INPUT (FIRST PAGE OF OUTPUT) HAS BEEN PRINTED FOR THIS CASE
```

```
C
      IF (NUMPAG.EQ.1)  GO TO 1000
C
C         THEN PRINT REGULAR OUTPUT HEADING LINES
         WRITE (6,9930)  TYPSUP(NF), (TYPWAV(I,NWAVEF), I=1,2),
     1                   (TYPSLB(I,NAIDA), I=1,4), TYPLOA(LOADFG)
         WRITE (6,9910)
         GO TO 2000
C
C         ELSE PRINT INPUT HEADING LINE
 1000    WRITE (6,9920)
C
C         END IF (BEG OF 1ST PAGE)
C
C         RESET LINE COUNT
C
 2000 NUMLIN = 12
      RETURN
C
C
C         FORMAT STATEMENTS
C
 9900 FORMAT (1H1/1H0/22X, *CALCULATIONS ON A CONCRETE *, A5, 50X,
     1        A10, 4X, A9/7X, 15A5, 43X, *PAGE*, I3)
 9910 FORMAT (1H0/ 5X, *TIME*,
     1               10X, *THETA*,
     2               10X, *MIDPT. VEL.*,
     3               4X, *MIDPT. DELTA*,
     4               4X, *PRESSURE WORK*,
     5               3X, *PLASTIC WORK*,
     6               3X, *KINETIC ENERGY*,
     7               3X, *HINGE LOCATION*/
     8               3X, *(SECONDS)*,
     9               5X, *(RADIANS)*,
     A               9X, *(IN./SEC.)*,
     B               6X, *(INCHES)*,
     C               1X, 3( 6X, *(IN.-LBS.)* ),
     D               7X, *(INCHES)* / )
 9920 FORMAT (1H0/1H0, 2X, *INPUT VALUES*)
 9930 FORMAT (1H0, 10X, A7, *-SUPPORTED*, 10X, 2A6, * FUNCTION*,
     1        10X, 4A8, 10X, A6, * LOAD*)
 9940 FORMAT (1H0, 20X, *AN ASTERISK INDICATES THAT A REINFORCING *,
     1              *ELEMENT HAS FRACTURED*)
C
      END
```